Adoptive governance in distributed agile development projects

Adoptive Governance in Distributed Agile Development Projects

Esmeralda Thomson

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Business

Faculty of Information Systems and Technology

July 2017

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: Thomson

First name: Esmeralda

Other name/s:

Abbreviation for degree as given in the University calendar: PhD

School: School of Business

Faculty: Information systems & Technology Management

Title: Adoptive Governance in Distributed Agile Development Projects

The current study responds to calls for research to increase the knowledge on how development firms could generate and enhance value from their distributed IT projects. Recent literature indicates the need for studies to address significant challenges of distributed information systems development teams, such as cultural incompatibility, lack of trust, customer collaboration, communication, lack of control and coordination (Mattsson et al. 2010).

Based on a qualitative exploratory single-case study, this dissertation is concerned with the institution of effective governance frameworks to address distributed development project challenges, an area of research that is currently lacking empirical studies. In particular, the current study is seeking to understand how social governance mechanisms affect the governance of distributed software development projects.

The current study shows the role and intervening processes of social governance mechanisms (Jones et al. 1997), including restricted access, macroculture, collective sanctions and reputation, to coordinate activities and safeguard exchanges. Furthermore, the current study suggests that to apply these social governance mechanisms effectively, it is critical that organisations maintain congruency among them. The study also found that all the four social governance mechanisms of the Jones et al. (1997) model interact with each other, thus showing the critical importance of macroculture among the social governance mechanisms and the impact that macroculture has on other mechanisms.

Moreover, the current study found that the mentioned four social governance mechanisms are context dependent and have different impacts on safeguarding and coordinating exchanges in various contexts including Open Source Software. The current study also found a new construct, ‘ideological similarity’, which is about a preference for more frequent interactions among project teams with similar interests to facilitate smooth interactions and enhance coordination. The study also provides effective practices, such as co-locating distributed teams at the start of the project and through the project life cycle.

FOR OFFICE USE ONLY	Date of completion of requirements for Award:

ORIGINALITY STATEMENT

‘I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.’

Signed ……………………………………………..............

Date ……………………………………………..............

COPYRIGHT STATEMENT

Signed ……………………………………………..........................

Date ……………………………………………..........................

AUTHENTICITY STATEMENT

‘I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.’

Signed ……………………………………………..........................

Date ……………………………………………..........................

ACKNOWLEDGEMENT

The PhD was an exciting academic journey that involved the most intense intellectual and emotional engagement I have had in my academic life. It shaped me and mould me in different ways. I feel so fortunate for having this opportunity to experience it at one stage of my life.

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Ken Steven, who has supported me throughout my thesis with his patience and knowledge. I am extremely grateful for his valuable guidance, scholarly inputs and consistent encouragement I received throughout the research work. I could not have imagined having a better advisor and mentor for my PhD study. My thesis would not have come to a successful completion, without your help and support.

I would also like to express my appreciation to Professor Richard Vidgen and Professor Kieran Conboy for their enthusiasm, enlightening guidance and support throughout my research journey, that have helped me in completing this research. I would like to thank Prof. Walter Fernandez for his constructive comments in final stage of my research. Many thanks go to Professor Patrick Finnegan for his lectures, support and encouragement which gave me in depth knowledge and understanding.

I would like to thank the editors and anonymous reviewers of the published articles associated with this thesis for their precious time in reviewing my works and for their valuable comments and suggestions. A special thank you to the people who helped me during the data collection process.

A special gratitude and love goes to my family for their unfailing support, I attribute my achievements to them. I would like to thank my amazing mother and my life coach, Lily Thomson, for her endless love and support. She always stood by me through life's ups and downs and encouraged me with her best wishes. I would like to thank my darling son, Edward William, for his sweet smile, for being patient with me when I did not have time to play and for his beautiful presence that has added much shine and love to my life.

Above all, I owe it all to Almighty God for granting me the wisdom, health and strength to undertake this research task and enabling me to its completion.

ABSTRACT

The current study responds to calls for research to increase the knowledge on how development firms could generate and enhance value from their distributed IT projects. Recent literature indicates the need for studies to address significant challenges of distributed information systems development teams, such as cultural incompatibility, lack of trust, customer collaboration, communication, lack of control and coordination (Mattsson et al. 2010).

Based on a qualitative exploratory single-case study, this dissertation is concerned with the institution of effective governance frameworks to address distributed development project challenges, an area of research that is currently lacking empirical studies. In particular, the current study is seeking to understand how social governance mechanisms affect the governance of distributed software development projects.

The current study shows the role and intervening processes of social governance mechanisms (Jones et al. 1997), including restricted access, macroculture, collective sanctions and reputation, to coordinate activities and safeguard exchanges. Furthermore, the current study suggests that to apply these social governance mechanisms effectively, it is critical that organisations maintain congruency among them. The study also found that all the four social governance mechanisms of the Jones et al. (1997) model interact with each other, thus showing the critical importance of macroculture among the social governance mechanisms and the impact that macroculture has on other mechanisms.

Moreover, the current study found that the mentioned four social governance mechanisms are context dependent and have different impacts on safeguarding and coordinating exchanges in various contexts including Open Source Software. The current study also found a new construct, ‘ideological similarity’, which is about a preference for more frequent interactions among project teams with similar interests to facilitate smooth interactions and enhance coordination. The study also provides effective practices, such as co-locating distributed teams at the start of the project and through the project life cycle.

Table of Contents

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

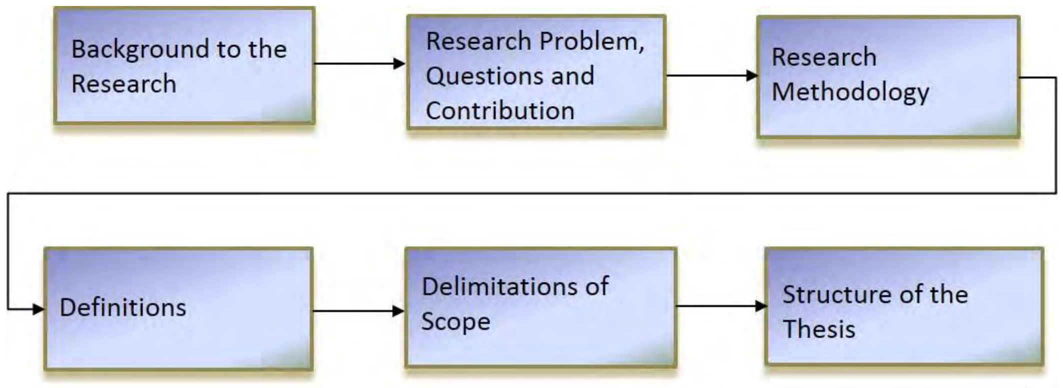
Table **Page**

Chapter 1 Introduction

1.1 Introduction

The purpose of Chapter 1 is to introduce this study. This Chapter starts by providing a background to the research. Then the Chapter explains the research problem, objectives, questions and contribution. Subsequently a summary of the research methodology is introduced followed by definitions of terms used in the study. Next delimitations of the research scope are provided. Finally the thesis structure is presented. Figure 1.1 summarises the structure of this Chapter.

Figure 1. 1 The Structure of Chapter 1



1.2 Background to the Research

Today, software development projects have increasingly moved away from the traditional co-located model, where teams are in one physical location, towards the off-shore model in which distributed teams collaborate from multiple places. Distributed Development (DD) arrangements mean these projects often suffer. Developing across geographical distance intensifies challenges like peer communications, shared understanding between teams and systems integration (Ågerfalk & Fitzgerald 2009, Sangwan *et al.* 2007). This increasing movement toward distributed software development and the formation of distributed teams poses a potential dilemma for organisations who have adopted *agile* software development methods

(Hildenbrand *et al.* 2008). These organisations face challenges because agile principles state that one of the most efficient and effective ways of conveying information to and within a development team is face-to-face conversation, where teams are co-located, which is not practical in distributed context (Beck *et al.* 2001).

Due to the physical separation of development teams in a distributed environment, many of the key assumptions within agile development, such as customer interaction and co-located teams with daily face-to-face communication, are not applicable (Dingsøyr *et al*. 2012). Indeed, within the context of distributed software development, the teams must rely on technology tools to communicate and in some cases they never meet in person (Ramesh *et al.* 2006).

Moreover, while agile methods mainly rely on social (informal) processes to facilitate coordination, distributed software development typically relies on formal mechanisms. The current literature has mostly focussed on the challenges of agile software development in a distributed context such as cultural incompatibility, lack of trust, control, communication and collaboration (i.e. Mattsson *et al.* 2010, Passivaara & Lassenius 2006, Ramesh *et al.* 2006). To address these challenges, many researchers propose that firms combine the flexibility offered by the agile development approach with software development in a distributed context, attempting to combine the two into a common distributed agile development model (Ågerfalk & Fitzgerald 2009; Ramesh *et al.* 2006). According to the study conducted by Hildenbrand *et al.* (2008), agile software development projects face difficulties in their implementations and executions, especially in a distributed environment. For example, customer involvement in the development process becomes difficult; intense involvement between the developers that is typical for agile development projects may suffer from cultural differences within the team; communication becomes challenging because of the distance. Hence, it is critical to devise solutions in which flexibility and agility values are preserved in DD as much as possible.

In recent years there has been growing concern about how firms can align their IT organisation with their business and successfully extract value from their IT investments. Effective governance has been identified as the single most important predictor of the value that an organisation generates from its IT activities (Weill & Ross 2004). The study conducted by Weill and Ross (2004) found that, in general, organisations with effective IT governance programs gained twenty percent higher profit margins than those organisations with poor quality governance programs that had similar strategic goals. Therefore, it is also important to devise solutions to distributed agile development challenges that provide a framework that enables development firms to benefit from their IT activities.

The effective agile governance facilitates business value, improved performance, monitoring and then control of large agile software development environments by aligning business goals and agile software development goals (Qumer 2007). Thus, to benefit from their IT activities, development firms require an effective governance framework with a set of control mechanisms to address DD project challenges. The transition from a traditional plan-based development approach to agile software development applied in a DD environment involves challenges and suggests the critical need for an effective governance framework to address these challenges.

1.3 Research Problem, Research Questions and Research Contribution

The DD method has received increasing interest in literature as the software community invites the scholars to find new solutions to DD challenges. In order to investigate the challenges of DD projects, researchers have looked into different dimensions. For example, Ramesh *et al.* (2006) in their study of three development organisations suggest that distributed software development can be agile when some unique characteristics of both environments are successfully blended. The study suggests five practices that are essential to achieve the balance between distributed and agility and overcome the challenges involved: continuously adjust the process; facilitate knowledge sharing; improve communication; build trust; and trust but verify. The study concludes that careful consideration of agility in distributed software development environments is essential when addressing challenges to communication, control and trust across DD project teams.

The transition from ‘heavyweight’ or ‘rigid’ development methodologies can present implementation issues, yet it is important that organisations assess how this transition can be executed in their own context (Boehm 2002). In the same line of argument, the literature suggests the creation of an ‘agile-rigid’ environment, where the formal structures of the traditional approach, such as development of a project plan and communication plan, applies (Yadav *et al.* 2007). The advantage of the ‘agile-rigid’ environment is combining the rigour of traditional overarching development approaches with the flexibility of agile approaches implemented within the project teams (Meso & Jain 2006, Yadav *et al.* 2007). It can also help organisations mitigate communication, coordination and control related risks inherent in globally distributed software development (Yadav *et al.* 2007).

As the interest in using agile approaches in DD projects is growing; so is the research literature on various mechanisms (Hossain *et al.* 2009), challenges and strategies of deploying the agile development approach for distributed projects (Sureshchandra & Shrinivasavadhani 2008, Young & Terashima 2008, Hildenbrand *et al.* 2008, Sutherland *et al.* 2007). Nevertheless, research on distributed agile development is scarce (Passivaara *et al.* 2009). There are only a few reported experiences in applying distributed agile development to industrial projects and even fewer case studies available in literature, such as Passivaara *et al.'s* (2009) multiple case study of applying Scrum in global software development project and Sureshchandra and Shrinivasavadhani's (2008) case study of distributed agile development project. However, as distributed agile development becomes a fact of life in many companies, the organisations are becoming more interested in using distributed agile development, or in some cases have already started to use it (Sutherland *et al.* 2007). However, there has not been any significant effort to suggest any mechanisms as the solution to address these challenges and problems in literature.

Thus the problem for the current study is *"How can distributed development projects be governed effectively*?"

To address the current research problem, this study focussed on exploring how governance mechanisms can resolve challenges such as coordination and safeguarding in distributed agile development projects. The theoretical aim of this research was developing a model of governance that could contribute to ensuring IT benefits are achieved in DD projects. The model presented in this dissertation (see Chapter 6) effectively applies social governance mechanisms in DD projects that use agile methodology and practice free flow of information, coordination and flexibility.

The practical aim of this research was to explore the potential benefits and challenges of adopting social governance mechanisms in DD projects. In doing so, the current study addresses two important points in the literature related to governance of DD projects that guided this research:

1) As projects become more distributed the increasing complexity in their infrastructure requires an effective governance approach that enhances value of their IT activities;

2) Know-how of applying a governance approach that is compatible with agile values/social governance is necessary.

The first point regarding governance of DD projects is related to the challenges of development projects in a distributed context. The physical distance between the development teams means a reduction in coordination, face-to-face team interactions and on-site customer and co-located team collaboration (Yadav *et al.* 2007). This means the more a development project is distributed physically, the more challenges are involved. Therefore, the companies need to have the knowledge of how they can enhance the value of their IT activities (Pye and Warren 2006). The important point is that all of these complications cause more complexity in the system that call for effective governance.

The second point is about developing compatible and effective governance mechanisms for distributed agile development projects and having the knowledge of how to apply these mechanisms. Although distributed agile development projects require governance, the imposition of a top-down, central, formal governance that has a strict control nature is in conflict with bottom-up, flexibility and empowering characteristics of agile development projects (Qumer 2007). Hence, to have the compatible and effective governance for distributed agile development projects, it is critical to develop a governance approach that appreciates agile values and principles, flexibility and empowerment of the project teams in a DD environment as well as enables development organisations to benefit from their IT activities (Anderson & Carney 2009). The question is what governance mechanisms can facilitate this?

The primary research questions that guided the study are as follow:

RQ1) How are social governance mechanisms applied in distributed agile development projects?
RQ2) How do social governance mechanisms coordinate exchanges among distributed agile development project teams?

RQ3) How do social governance mechanisms safeguard exchanges among distributed agile development project teams?

This research aim to make a significant contribution to both IS researchers and IT practitioners by investigating different facets of governance mechanisms in DD projects. The contribution of this study for IS researchers is to the knowledge base on how software development projects can be governed effectively in a distributed context. The current study provides an empirical support for the Network Governance (NG) theoretical model developed by Jones *et al.* (1997) and contributes to the existing model as a useful framework for studying governance mechanisms in distributed projects. It presents a good description of social governance that can be used in DD projects.

The current study provides practical implications that can be used by practitioners and a major contribution to practitioners and IS researchers by providing a list of effective practices for governance of development projects in a distributed environment. This research also provides a basis for future research on how DD projects can be governed using theory as a NG theoretical lens. Having a valid and reliable model of governance allows the empirical exploration of different facets of governance in DD projects with a solid theoretical core.

1.2 Research Methodology

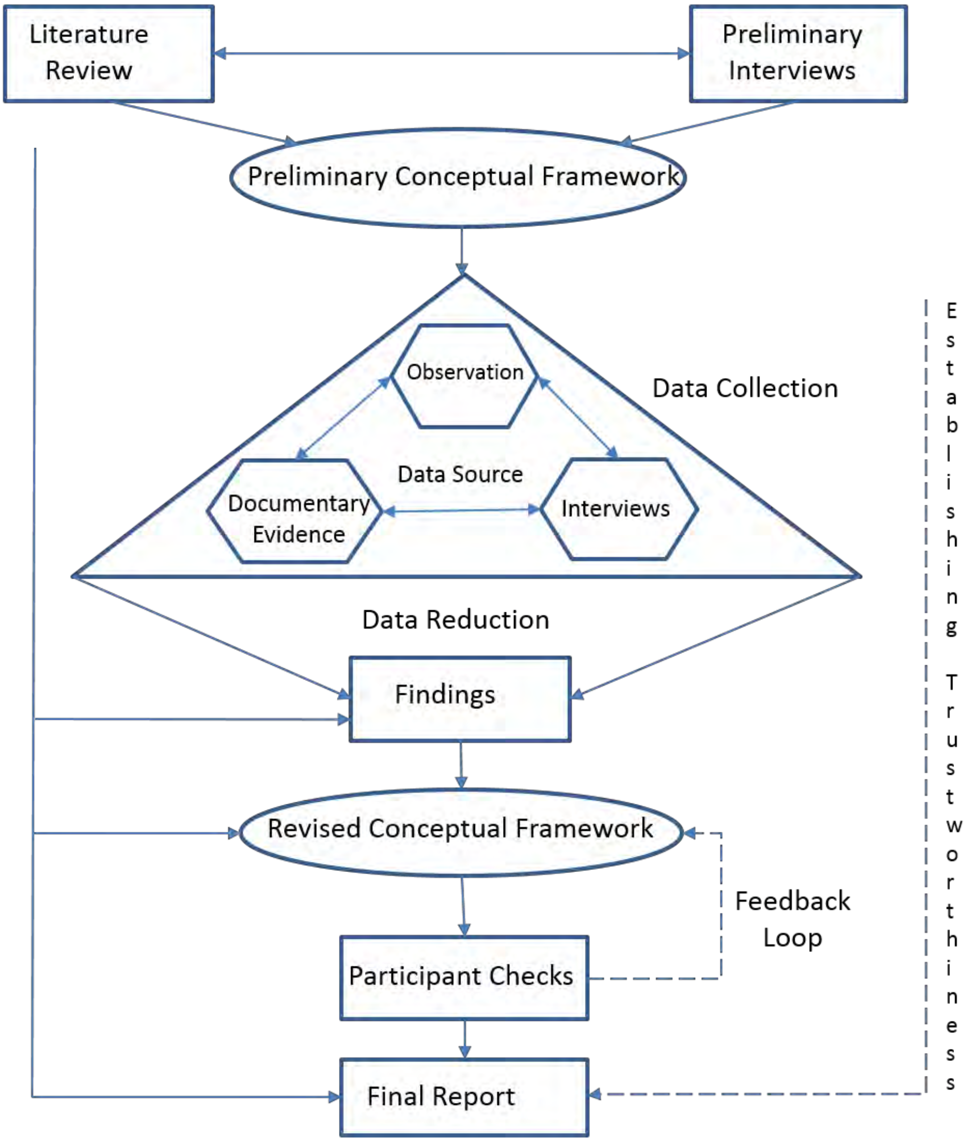
The current study aimed to explore and develop a governance approach for distributed agile development projects in an original manner. Therefore, in order to have a detailed contextual analysis of how social governance mechanisms are applied in distributed context and by considering the nature of the study, the research problem and theoretical framework used, I adopted a qualitative approach with interpretivist ontology (Creswell 2007). In addition, I chose a qualitative research method because understanding of the participants being studied was highly significant in extending the knowledge base about application of social governance mechanisms and their impact on coordinating and safeguarding of exchanges in distributed agile development projects.

An exploratory case study is particularly appropriate when the field of study is still evolving (Walsham 1993). The case study method allows in-depth investigation to be conducted by retaining the holistic and meaningful characteristics of real-life events (Yin 2003). Therefore, I chose a case study design. The case study approach allowed for a detailed contextual analysis of distributed agile development projects' social interaction and underlying governance mechanisms.

The case project chosen for the current study is referred to as 'Lake Project'. The Lake Project was the first distributed agile project in the organisation and considered a high profile project. The Lake Project was distributed geographically across Australia (Sydney and Melbourne) and China (Chengdu). Agile methodology was adopted organisation-wide and was part of the development norm. The Lake Project was selected as the suitable case study for this research because it was considered a high profile project that was distributed across locations with very well established agile methodology (Section 4.2.4). In addition, the Lake project met the selection criteria required for NG theory to emerge (Section 4.3).

The current research was interested not only in studying and describing the observable behaviour of the respondents but also aimed to understand their values, norms, interests, motivations. Such specifications made the study more compatible with interpretivist ontology (Walsham 2006). The research design is summarised in Figure 1.2. The research methodology is discussed in more details in Chapter 3.

Figure 1. 2 Research Design



1.5 Definition of Terms

For clearer understanding of the terms used in the current study, below are their meanings:

***Development**:* for the purpose of the current study, I chose to define “development” broadly as “any software development lifecycle activity”. This extends beyond “pure” development activities and includes, for example, deployment and maintenance. The study used the term “activity” in a loose sense, including any individual or collective human action at any level of granularity that serves a particular purpose. According to activity theorist Engeström (2000), an activity is something that transforms an object to an outcome. Hence, a “development activity” is that individual or collective action that transforms something abstract or concrete into something meaningful in the context of a software system's lifecycle. Thus, the current study would regard an individual developer's creation of a source code document as a development activity that transforms a requirements document into a piece of code.

Software Development: Software is developed with hundreds or thousands of people in teams. They perform a multitude of different activities, so called processes. Software development is the process of writing and maintaining source code, documenting, testing, and bug fixing with the intent of creating the desired software product in a planned and structured process (Münch *et al*. 2012).

Distributed development project*:* in the current study, classifying a project or development team as distributed means teams are spread out geographically across multiple locations, where the project teams may not see each other face to face, but they are all working collaboratively toward the outcome of the project (Gentleman *et al.* 2004).

Agile software development: In the current study agile software development is defined as a group of software development methodologies based on similar principles that promote a project management approach with frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organisation and accountability, a set of effective practices that allow for rapid delivery of high-quality software and a business approach that aims to align development with customer needs and organisation goals (Schwaber & Beedle 2002).

Governance is defined as the exercise of control and direction over a subject such as a society, an organisation, processes or artefacts, by using laws and policies that are defined, deployed and executed (Dubinsky & Kruchten 2009). This refers to the ***formal governance*** that relates to the command and control structures which are more common in traditional (top-down) software development approach.

Social governance is defined as an informal structure that is characterised by coordination of the organisation members' interactions, which are common in agile (bottom-up) software development approach (Jones *et al.* 1997).

Methodology*:* the current study refers to methodology as a system of broad principles or rules from which specific methods or procedures may be derived to interpret or solve different problems (businessdictionary.com 2001).

1.6 Delimitation of scope

Jones *et al.*'s (1997) model consists of four social governance mechanisms, including restricting access to exchanges, using collective sanctions, utilisation of macroculture and the importance of reputation to the project team members. The adopted model proposes that restricted access to the DD team projects, effective use of macroculture, collective sanctions and the importance of reputation to the project members lead to better coordination of efforts and better safeguarding of exchanges among the project teams. The theoretical model is summarised in Chapter 6, Figure 6.3.

The aim of the research was to understand how DD projects can be governed effectively. This research only focussed on social governance mechanisms and was not interested in formal governance mechanisms. I did not consider the interactions between social and formal governance. For the purpose of this research, I only focussed on identifying and assessing the application and effectiveness of the social governance mechanisms in the Jones *et al.* (1997) model. The current study identified the relationship among the research model social governance mechanisms but the in-depth analysis of these relationships was outside the scope of this research.

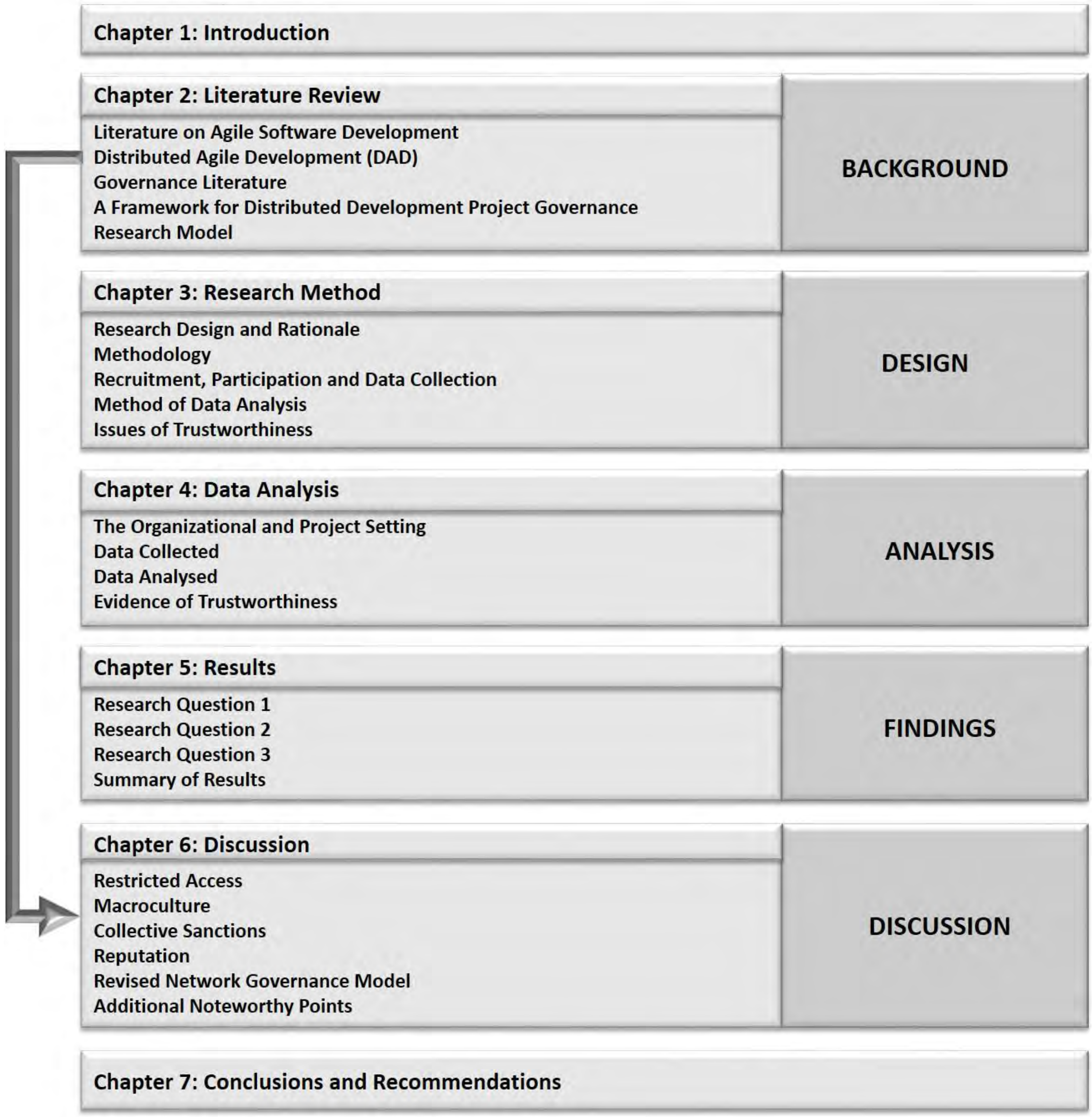
The model of social governance proposed by Jones *et al.* (1997) was adopted for this study. Jones *et al.* (1997) integrated Transaction Cost Eonomics (TCE) analysis and social network theory in order to explain why NG emerges and thrives. For a governance form to emerge and thrive, it must address problems of coordinating and safeguarding exchanges more efficiently than other governance forms (Williamson 1991). There are four exchange conditions necessary for social governance to emerge and thrive effectively (see Chapter 2): demand uncertainty, customised (asset-specific) exchanges, complex tasks executed under time pressure and frequent exchanges between the project teams. Jones *et al.* (1997) suggest that the social governance mechanisms provide comparative advantage over other governance forms for these exchange conditions. In order to be able to apply the Jones *et al.* (1997) model of social governance, this research covered a single case study that was compatible with the NG theory exchange conditions (see Chapter 5).

I only looked at distribution in the notion of different software elements being built by different teams at distributed geographical locations. I did not consider other forms of DD projects. In addition, I only focused on agile distributed software development projects. The reason for choosing an agile development project case was that agile software development projects meet the four exchange conditions required for social governance to emerge. Other IT projects, co-located development projects and other industry projects are outside the scope of this research.

1.7 Structure of the Thesis

The current study begins by providing an introduction explaining the research objectives and questions followed by research problem and motivation. Subsequently, Chapter 2 provides background about the relevant literature on agile software development, distributed agile development and governance followed by a detailed explanation of NG theory and presents the research model. Chapter 3 outlines the research design, the theoretical method, case study design justification as well as the research questions. Chapter 3 also discusses the research methodology, data collection and data management and analysis methods followed by issues of trustworthiness. Chapter 4 provides a detailed description about project settings, the data collected and analysed, followed by assurance of trustworthiness of these procedures. Subsequently, Chapter 5 presents the findings from themes that emerged during data analysis. Chapter 6 then discusses a summary of main findings, interprets and explains the findings and discusses how the results relate to the literature. The findings are reviewed in relation to the literature analysed in Chapter 2. Finally, Chapter 6 provides answers to the research questions and the significance and importance of the results, with reference to the literature. Subsequently, the chapter provides an additional discussion about the research's noteworthy points of what appeared to be useful for the practitioners. The study concludes with a summary of the research in Chapter 7, followed by a discussion of the study's limitations, the study's contributions to theory and implications for practice, then recommendations for future research and a conclusion. The thesis structure is summarised below in Figure 1.3.

Figure 1. 3 Thesis Structure


Chapter 1: Introduction
Chapter 2: Literature Review
Literature on Agile Software Development
Distributed Agile Development (DAD)
Governance Literature
A Framework for Distributed Development Project Governance
Research Model
BACKGROUND
Chapter 3: Research Method
Research Design and Rationale
Methodology
Recruitment, Participation and Data Collection
Method of Data Analysis
Issues of Trustworthiness
DESIGN
Chapter 4: Data Analysis
The Organizational and Project Setting
Data Collected
Data Analysed
Evidence of Trustworthiness
ANALYSIS
Chapter 5: Results
Research Question 1
Research Question 2
Research Question 3
Summary of Results
FINDINGS
Chapter 6: Discussion
Restricted Access
Macroculture
Collective Sanctions
Reputation
Revised Network Governance Model
Additional Noteworthy Points
DISCUSSION
Chapter 7: Conclusions and Recommendations

1.8 Conclusion

This chapter provided an introduction to the current research study by explaining the research problem, the research questions and the research contribution. This chapter also briefly described the research methodology, defined key terms, presented delimitation of scope and outlined the structure of this thesis. The following chapters provide a detailed description of the research and its outcomes.

Chapter 2 Literature Review

2.1 Introduction

The current study aimed to explore effective governance mechanisms being used in DD projects. The purpose of Chapter 2 is to outline the major streams of literature necessary for establishing the foundation of the study. Following Webster and Watson (2002) recommended literature review processes, this study (1) outlined past research and highlighted its gaps, (2) addressed these shortcomings by proposing use of a NG theoretical framework, (3) identified academics who have called for this research and (4) indicated that this research has important implications for practice.

A systematic and structured review of past literature is a crucial endeavour for any academic research (Webster & Watson 2002). The general purpose of conducting a literature review is to identify potential sources of information for conducting detailed research (Hart 1998). The literature review facilitates gaining background knowledge of the research topic, uncovering areas where research is needed, identifying problems that have already been investigated in the past and establishing a theoretical framework for the research topic (Webster & Watson 2002).

Webster and Watson (2002) defined an effective literature review as one that "creates a firm foundation for advancing knowledge. It facilitates theory development, closes areas where a plethora of research exists and uncovers areas where research is needed" (p. 13). I followed Webster and Watson's (2002) structured approach (outlined below) to identify the sources for the literature review. First, I identified major articles related to the subject under the study. Once an initial pool of papers were identified, I conducted backward searching. In "backward‶ searching, the citations for the articles that were identified in the first step, are then reviewed to discover previously written papers that would be relevant.

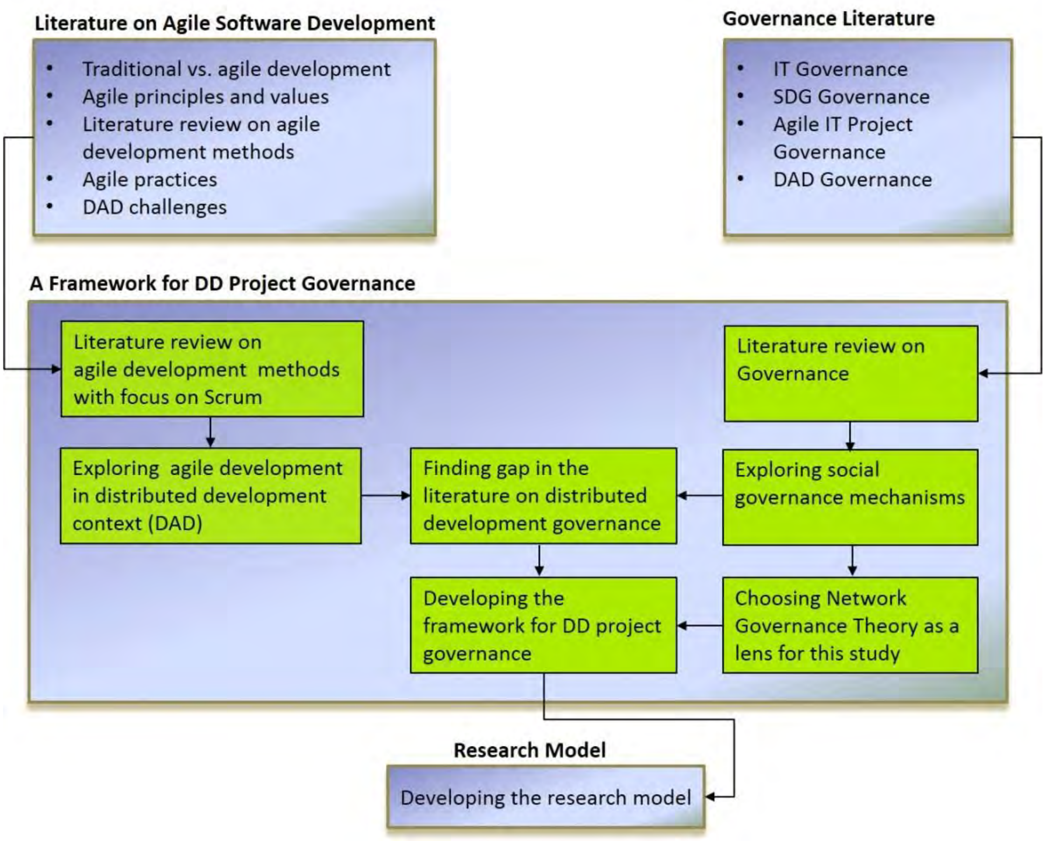
To identify major contributions, start with leading journals: I scanned conference proceedings and journals with reputation such as *Management Information Systems Quarterly (MISQ)* and *Information Systems Research*, to accelerate identification of relevant articles. I also examined well regarded conference proceedings such as International Conference on Software Engineering (ICSE), International Conference on Information Systems (ICIS) and European

Conference on Information Systems (ECIS). Further, I extended the review by not looking only within the IS discipline but also other fields such as business and management.

Go backward: After identifying the potential articles, I did a backward review by going through the citations of the identified articles to determine previous articles that needed to be considered. I conducted a thorough literature review of governance, outlined past studies, provided examples and highlighted the critical knowledge gaps in the literature. Further, I suggested covering the shortcomings by extending the current theory. Finally, I outlined the scholars who have called for this research and indicated that this research has important implications for practice.

This chapter reviews the relevant literature on agile software development and governance and reviews theories that were used to explain the research model. The first stream of literature relates to agile software development methods. The second stream reviews research in the areas of distributed agile development projects. The third stream of literature provides a review about governance in general and governance of distributed agile development projects in particular. Based upon these streams of literature, the conceptual model for the study was developed. Next, NG theory and its application to distributed agile development project teams were explained in details. Finally, the research model was developed using NG theory as a lens. Figure 2.1 summarises the structure of this chapter.

Figure 2. 1 The Structure of Chapter 2


Literature on Agile Software Development
Traditional vs. agile development
Agile principles and values
Literature review on agile development methods
Agile practices
DAD challenges
Governance Literature
IT Governance
SDG Governance
Agile IT Project Governance
DAD Governance
A Framework for DD Project Governance
Literature review on agile development methods with focus on Scrum
Literature review on Governance
Exploring agile development in distributed development context (DAD)
Finding gap in the literature on distributed development governance
Exploring social governance mechanisms
Developing the framework for DD project governance
Choosing Network Governance Theory as a lens for this study
Research Model
Developing the research model

2.2 Literature on Agile Software Development

The majority of the software development processes were being criticised as bureaucratic, slow and overly regimented (Schwaber & Beedle 2002). Many early attempts to improve software development focussed on better ways of defining and detailing requirements, designing comprehensive architectures to support the requirements and then developing the software in a much more regimented, methodical manner to realise the system and supporting architecture (Baskerville *et al.* 2001). However, these attempts did not go far enough in addressing the needs of effectively managing rapidly changing requirements nor in accelerating the delivery of software (Abrahamsson *et al.* 2002, Middleton 2001, Dyba & Dingsoyr 2008).

Finally in the mid-1990s, in reaction to these *heavyweight* software methods, agile software development emerged as an innovative approach to software, enabling development organisations to quickly react and adapt to changing requirements and technologies that resulted in a more successful development strategy (Beck *et al.* 2001). *Agile software development methods* is referred to as an answer to the business community asking for lighter weight along with faster software development processes with the "ability to create and respond to change" (Highsmith & Cockburn, 2001 p. 120).

Indeed, software development organisations operate in a complex, uncertain and demanding environment with frequent interactions between different project members (Schwaber & Beedle 2002). In order to produce highly customised software products, these organisations are required to create value-added deliverables in a more effective and efficient way, while controlling both risks and budget (Dubinsky *et al.* 2008). Therefore, agile software development arose as a response to the following conditions (Baskerville *et al.* 2001):

- complexity of the development process
- the frequency of interaction between participants in the development process
- increased demand uncertainty of modern software development projects, which are often undertaken under conditions of intense pressure to deliver and turbulent business environments
- changing customer requirements
- rapidly changing technology

This changing and uncertain environment in software development organisations also affects the software development process to the extent that satisfying the customers at the time of delivery has taken precedence over satisfying the customer at the beginning of the project (Schwaber & Beedle 2002). Agile software development is an invitation for procedures that are not so much focussed on how to stop change in the early stages of a project, but on how to better handle the inevitable changes that arise throughout its lifecycle (Highsmith & Cockburn 2001). Agile software development methods are designed to respond to changes through the development process, therefore agile development methods are much more capable of satisfying the customers at the time of delivery (Yu & Petter 2014, Baskerville *et al.* 2001).

The Agile Manifesto identified the following central values (Beck *et al.* 2001):

1. Individuals and interactions are preferred over processes and tools; this value addresses the importance of direct communication among developers and the human role

reflected in the contracts, as opposed to institutionalised process and development tools

2. Working software is preferred over comprehensive documentation; this value insists that developers keep the code simple and technically as advanced as possible to reduce the documentation load
3. Customer collaboration is preferred over contract negotiation; this value emphasis continued relationship and cooperation between the developers and the client as opposed to strict contracts relationship
4. Responding to change is preferred over following a plan; this emphasis is on agile software development project teams comprising both software developers and customer representatives, who should be well-informed and have the authority to consider possible changes needs to be applied during development life cycle.

Ever since the agile manifesto was created in 2001, the research community has devoted a great deal of attention to agile software development. The early research on agile software development focussed on issues related to the adoption of agile methods (e.g. Boehm 2002 and Nerur *et al.*, 2005) and on the efficiency of pair programming in software development (Nawrocki & Wojciechowski 2001 and Williams *et al.* 2000). Other studies have investigated various aspects of team dynamics, e.g., trust, self-organisation, and communication (Moe *et al.* 2009), consequences of test-driven development (Erdogmus *et al.* 2005 and Janzen and Saiedian 2005), adoption and post-adoption issues (Cao *et al.* 2009 and Mangalaraj *et al.* 2009), challenges of implementing agile in distributed settings (Ramesh *et al.* 2006) and the like. Despite the extensive research on agile software development and its complex divisions, clearly more work has to be done because there is still a lack of unified framework that brings consistency to the different branches of research conducted (Dingsøyr *et al.* 2012).

The following sub-sections provide a comparison between traditional and agile software development method, followed by a discussion on XP and Scrum agile methodology. Subsequently, section 2.3 discuss the agile methodology in DD context and its challenges in turn.

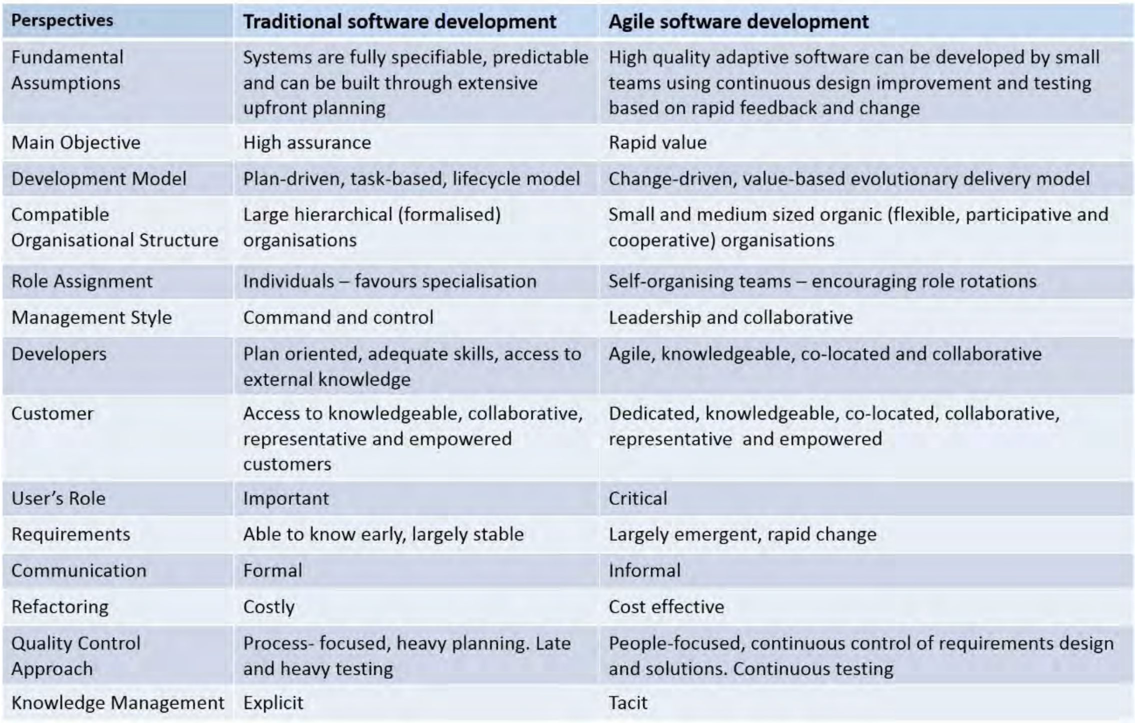
2.2.1 Traditional vs. Agile Development Method

Software development life cycle is either iterative or sequential. The agile software development is a distinct counterpoint and response to 'traditional' modes of software development (Lewis 2016). Traditional development attempts to minimise change during the project life cycle through rigorous upfront requirements such as information gathering, analysis and design (Boehm 2002). Indeed, the traditional lifecycle methodology tend to develop applications by performing big up-front application design first (Lewis 2016). The intention is to maintain higher quality results under a controlled schedule (Abrahamsson *et al.* 2002). In traditional software development, developers are assigned to specific roles related to the specific phase of the project. Some authors also find the documentation in traditional development to be excessive (Ambler 2009). Making any changes is very challenging and requires the approval from higher level management (Boehm 2002). Such hierarchical approach is not compatible with uncertain and changing customer requirements (Ambler 2009).

Unlike traditional software development methods, agile methods are marked by extensive collaboration, i.e. face-to-face communication (Inayat *et al.* 2015). In agile software development projects, there is rapid customer satisfaction by continuous delivery of useful software because working software is delivered within weeks rather than months and this working software is the principal measure of project progress (Yu & Petter 2014, Boehm 2002). In agile software development even late changes in requirements are welcomed and can be reflected in the project because close and daily cooperation between the customer and developers provides an easier approach for regular adaptation to changing circumstances (Beck *et al.* 2001). Overall, a high level project scope is defined upfront and is reviewed in each iteration. Then, requirements are initially identified and confirmed with the customer and listed in a customer requirements list; every couple of weeks they are discussed (e.g. in the Scrum method) and reprioritised to define the scope of the next iteration (Inayat *et al.* 2015).

Table 2.1 provides a summary comparison between the founding principles of agile software development compared with traditional plan-driven development approach.

Table 2. 1 Traditional vs. Agile Software Development



Perspectives	Traditional software development	Agile software development
Fundamental Assumptions	Systems are fully specifiable, predictable and can be built through extensive upfront planning	High quality adaptive software can be developed by small teams using continuous design improvement and testing based on rapid feedback and change
Main Objective	High assurance	Rapid value
Development Model	Plan-driven, task-based, lifecycle model	Change-driven, value-based evolutionary delivery model
Compatible Organisational Structure	Large hierarchical (formalised) organisations	Small and medium sized organic (flexible, participative and cooperative) organisations
Role Assignment	Individuals – favours specialisation	Self-organising teams – encouraging role rotations
Management Style	Command and control	Leadership and collaborative
Developers	Plan oriented, adequate skills, access to external knowledge	Agile, knowledgeable, co-located and collaborative
Customer	Access to knowledgeable, collaborative, representative and empowered customers	Dedicated, knowledgeable, co-located, collaborative, representative and empowered
User's Role	Important	Critical
Requirements	Able to know early, largely stable	Largely emergent, rapid change
Communication	Formal	Informal
Refactoring	Costly	Cost effective
Quality Control Approach	Process- focused, heavy planning. Late and heavy testing	People-focused, continuous control of requirements design and solutions. Continuous testing
Knowledge Management	Explicit	Tacit

Adapted from Nerur and Balijepally (2007), Boehm (2002).

The incremental, cooperative (customer and developers working together constantly), straightforward (easy to learn and to modify, well documented) and adaptive practices make a software development method an agile one (Abrahamsson *et al.* 2002). These principles facilitate close interaction between developers and customer representatives that result in increased trust and a better understanding (Beck and Andres 2004, Schwaber & Beedle 2002). The principles are made concrete through the following practices (Beck & Andres 2004):

- co-locating users and developers to improve communication
- pair programming; when two developers work in front of a single computer, coding and testing the software together in pair, which is a great facilitator of communication within the pair and sharing of tacit knowledge
- short development iterations (typically from one week to one month) to rapidly reflect customer demands and changing requirements
- daily Stand up meetings to review progress and facilitate communication among the entire team by providing visibility of individual's work to the rest of the team and among the team members who may not talk to each other regularly

- planning game to prioritise user requirements and the business domain between customer representatives and the developers

An agile software development team operates as an independent work unit embedded in a larger enterprise (and possibly an inter-organisational setting) in an environment that encompasses many stakeholders, including IT developers and testers, business managers and customers. The IT and business stakeholders need to work together as a single entity, or network, to achieve business value for the enterprise (Sutherland *et al.* 2007). The 'customer' is a special type of business stakeholder; they are the business owner of the project and are accountable for the value created by the project.

Over the course of a decade, agile software development has gone from being a controversial practice to 'business as usual' for many firms. Ever since the manifesto was articulated, practitioners and researchers have been trying to explicate agility and its different facets. More formal definitions of agility have started to appear in the recent past, drawn mainly from manufacturing and management domains, where agile appears to have its roots. For Henderson-Sellers and Serour (2005), agility involves both the ability to adapt to changes and to refine and fine-tune development processes as needed. Lee and Xia (2010) define software development agility "as the software team's capability to efficiently and effectively respond to and incorporate user requirement changes during the project life cycle". Conboy (2009) provides by far the most comprehensive definition of software development agility by systematically examining its various facets and definitions from related disciplines.

A variety of agile methods, such as Scrum (Schwaber & Beedle 2002) and eXtreme Programming (XP) (Beck & Andres 2004), have become particularly popular. Scrum is the most widely adopted iterative and incremental software development approach that focusses on day to day project management (Schwaber & Beedle 2002) and is the most popular agile approach that has been recently studied in distributed agile development literature (i.e. Paasivaara *et al.* 2009, Hossain *et al.* 2009, and Sutherland *et al.* 2007). The next paragraphs discuss an overview of the most common agile development methods.

2.2.2 Extreme Programming (XP)

XP is a customer driven development method suitable for small teams with daily code builds (Yu & Petter 2014). It is the most documented of the other agile methods (Abrahamsson *et al.* 2002). The term ‘extreme’ comes from taking the XP practices into extreme level such as planning game, simple design, collective ownership and continuous integration (Beck & Andres 2004). However, the special feature of XP is ‘refactoring’ which is the ongoing redesign of the system to improve its performance and responsiveness to change.

XP phases

The life cycle of XP consist of different phases and the study explained them below according to Beck’s description (1999b):

1. Exploration: in this phase the customers write the user story cards or requirements that the. The development team converts user stories into iterations that cover a small part of the functionality or features required. A combination of iterations provides the customer with the final fully functional product.
2. Planning: this phase sets the priority order for the story cards. The programming team prepares the plan, time, and costs of carrying out the iterations, and individual developers sign up for iterations. The planning phase usually takes a couple of days.
3. Iterations to release: in this phase the schedule set in the planning phase is broke down into several iterations before the first release. This is achieved by selecting the story cards that will enforce building the structure for the whole system.
4. Productionising: is this phase, the system performance goes through additional testing and checking before it can be released to the customer. During this phase, any new changes and suggestions are documented for later implementation during, for example, the maintenance phase.
5. Maintenance: after the first release is productionised for customer use, the XP project must both kept the system in the production running while also producing new iterations. In order to do this, the maintenance phase may requires to add new members into the team for customer support tasks.

6. Death: in this phase the customer does not have any stories to be implemented but the system should satisfy the customer needs in terms of performance and reliability. This is when in XP process the necessary documentation of the system is written as no more changes can be made.

XP Roles and responsibilities

There are different roles in XP for different tasks and purposes during the XP process. In the following, these roles are presented according to Dubinsky & Hazzan (2004) and Abrahamsson *et al.* (2002):

Programmer: writes tests, estimates stories, estimates how long stories and tasks will take, implements stories and keep the program code as simple and definite as possible.

Customer: writes user stories and specifies functional tests, set priorities and explain stories.

Testers: implement and run functional tests on regular basis. They broadcast test results and make sure people know when test results decline.

Tracker: gives feedback and traces the estimates made by the tem. He listens to the programmers, take action if things seem to be going off track. He also traces the progress of each iteration to assess whether the progress and goal of each iteration is achievable within the estimated time or of any changes is required.

Coach: watches everything as he is responsible for the whole process. He sends obscure signals and makes sure the team members following the XP process.

Consultant: an external member that processes the specific technical knowledge in order to help the team resolving specific problems.

Manager: makes decisions and communicates with the project team, schedules meetings and makes sure the meeting process is followed, records results of meeting for future reporting.

Some of the XP practices that were used in the current case study more frequently are discussed according to Abrahamsson *et al.* (2002) in the following:

Pair programming: two programmers write the code together at one computer.

Collective ownership: anyone can make changes to the written codes at any time.

Continuous integration: a new piece of code is integrated into the code-base as soon as it is ready, this makes the system to be integrated many times a day. All tests are run and have to be passed for the changes in the code to be accepted.

Testing: Unit tests are implemented and run continuously.

Planning game: Close interaction between the customer and the programmers about the effort needed for the implementation of the customer stories and the cope and time of the releases.

2.2.3 Scrum

Scrum is probably the most popular of all the agile life cycles with emphasis on the teamwork (Jorgensen 2016). Scrum was designed to add energy, focus, clarity and transparency to development project teams and achieve a hyper-productive state (Schwaber & Beedle 2002). A properly implemented Scrum allows for rapid development, aligns individual and organisation objectives, creates a culture driven by performance, supports shareholder value creation, achieves stable and consistent communication of performance at all levels and enhances individual development and quality of life (Cockburn 2002).

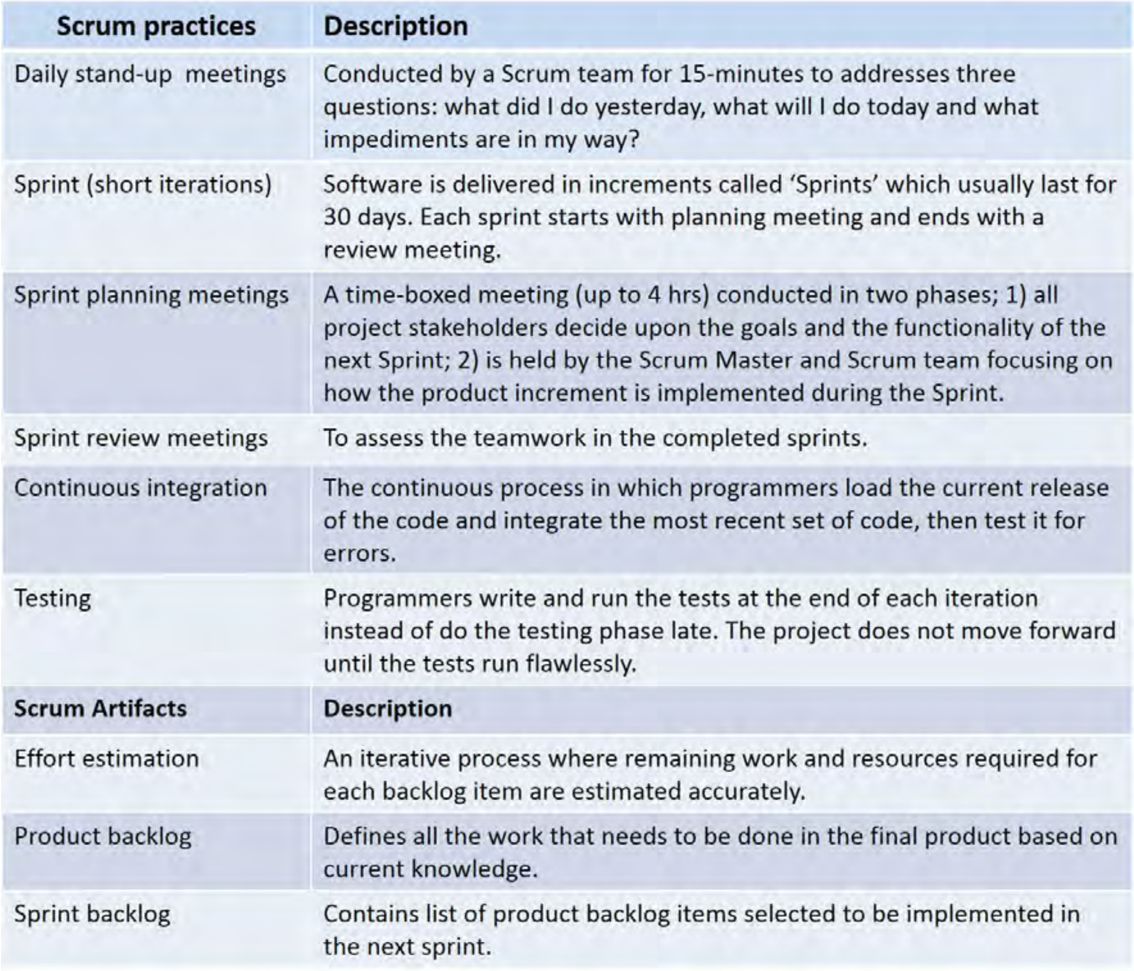
The main objective of Scrum is to manage the development project and be able to respond to unexpected changes during the development process. The Scrum process comprises three phases: pre-game; development (game phase); and post-game (Schwaber & Beedle 2002). The pre-game phase includes two sub-phases: planning (the definition of the system being developed); and architecture, where the high-level design of the system is planned. The development phase is the agile part of the Scrum where the unpredictable changes are expected, observed and controlled through different Scrum practices. The post-game phase is the closure of the release which includes integration, system testing and documentation. Although Scrum provides specific details about how to manage the 30 day release cycle, the integration and acceptance tests are not detailed (Abrahamsson *et al.* 2002).

Six roles are identified in Scrum (Schwaber & Beedle 2002):

- Scrum Master; responsible for ensuring that the project is carried through according to the practices, values and rules of Scrum and it progresses as planned
- Product Owner; responsible for the project, managing, controlling the product backlog list and making the final decision of the tasks related to product backlog
- Scrum Team; the project team that has the authority to decide on the necessary actions and to organise itself in order to achieve the goals of each Sprint
- Customer; participates in the tasks related to product backlog items
- Management; in charge of final decision making, along with the charters, standards and conventions to be followed in the project

Table 2.2 summarises Scrum common practices and artefacts (Schwaber & Beedle 2002) and some common XP practices (Cockburn 2002).

Table 2. 2 Scrum Practices



Scrum practices	Description
Daily stand-up meetings	Conducted by a Scrum team for 15-minutes to addresses three questions: what did I do yesterday, what will I do today and what impediments are in my way?
Sprint (short iterations)	Software is delivered in increments called 'Sprints' which usually last for 30 days. Each sprint starts with planning meeting and ends with a review meeting.
Sprint planning meetings	A time-boxed meeting (up to 4 hrs) conducted in two phases; 1) all project stakeholders decide upon the goals and the functionality of the next Sprint; 2) is held by the Scrum Master and Scrum team focusing on how the product increment is implemented during the Sprint.
Sprint review meetings	To assess the teamwork in the completed sprints.
Continuous integration	The continuous process in which programmers load the current release of the code and integrate the most recent set of code, then test it for errors.
Testing	Programmers write and run the tests at the end of each iteration instead of do the testing phase late. The project does not move forward until the tests run flawlessly.
Scrum Artifacts	**Description**
Effort estimation	An iterative process where remaining work and resources required for each backlog item are estimated accurately.
Product backlog	Defines all the work that needs to be done in the final product based on current knowledge.
Sprint backlog	Contains list of product backlog items selected to be implemented in the next sprint.

Scrum faces some challenges that are identified in the literature as follows:

- *Placing an overemphasis on the Scrum process and practices* (Marchenko & Abrahamsson 2008); the Scrum Master tends to be "too concerned" about solving the project teams' problems, instead of allowing them to solve the problems themselves. This can mean the project teams find the Scrum as too strict and against the common sense
- *The Scrum Master caring only about the individuals and interactions and ignoring the process* (Marchenko & Abrahamsson 2008)*;* this can cause the team to lose the feeling of discipline essential in software engineering
- *Difficulty in tracking progress and in using the results of the tracking* (Marchenko & Abrahamsson 2008)*;* unfortunately the results of the careful tracking take time to

become visible. This causes the project teams to become bored with the tracking process

- *Too much management interference* (*Marchenko & Abrahamsson 2008*); one of the Scrum's main components is the team's self-organisation. - it does not match traditional ways of developing software where management tries to interfere
- *Lack of clear accountability (Rayhan al. 2008);* a challenge for the development team to understand the context of self-organisation. It's easy to understand that each person is responsible for his or her part of the scope. However, in practice, it's challenging for the development team to understand how to be collectively responsible for the overall delivery of the work
- *Lack of agile experience (Hajjdiab et al. 2012)*; lack of experience with agile development methods or Scrum implementation in particular. The experience of traditional methods is completely different to committing to daily meetings, working with time boxes, finishing tasks in small period iteration and documenting the stories (or backlogs). Changing project management and programming habits in weeks or even months is difficult because it comes with training and more serious commitment to change
- *Planning meeting (Passivaara et al. 2009);* time-zone differences cause challenges to arrange meetings, especially longer ones. Cultural and language differences may cause the silence of some participants and it can be difficult to recognise speakers when not seeing the faces.

In their case study of implementing Scrum methodology, Fitzgerald *et al*. (2006), discussed some of the key advantages of Scrum, with them being:

- Planning and tracking become a collaboration involving the whole team
- Excellent communication builds up within the team, thus building morale and helping the team to be unified
- The team lead has more bandwidth for technical work
- Scrum enables the team to deliver on time
- The early adoption of Scrum leads to the formulation of internal training courses
- The use of Scrum leads to consistent meeting of development schedules on very complex projects with long project durations, but with no degradation in product quality

Further benefits of the Scrum methodology in distributed agile development context is discussed in the following section.

This section briefly discussed agile development methods, their characteristics and practices with an emphasis on Scrum. Agile methods are touted as a way to reduce development challenges and pave the way for the future of development (Beck & Andres 2004), however the increasing movement toward distributed software development poses a potential dilemma for organisations who have adopted agile methods (Ramesh *et al.* 2006). Most of the agile practices are applicable to co-located, small projects and adopting them in a DD environment can be challenging (Abrahamsson *et al.* 2002). The following section discusses the challenges that agile development projects are facing in a distributed context.

2.2.4 Kanban

Kanban uses cards or electronic signals in a computer system to monitor and control workflow. There are five qualities that Kanban has (Jyothi & Rao 2011):

1. Transparent workflow: Kanban makes the workflow visible to all employees.
2. Confines work in progress: it uses a pull procedure where a signal is sent to pull the material into the production area when it is needed as part of the manufacturing process.
3. Manages flow: It makes sure that the progress of function by way of each phase and aspect of workflow is observed, calculated and recorded.
4. Makes system strategies open: the visibility that Kanban provides, allows senior managers and employees to speak about new explicit policies, permitting them to understand more and to make improvements through a far more balanced way as all bases are put on record.
5. Allows mutual development: Kanban supports the practice of recognizing and applying little but concrete improvements that ends in evolutionary amendments.

Following are some of the advantages of Kanban method implementation according to Poppendieck and Poppendieck (2003)

- Creates a more efficient, effective operation and quality improvement.

- Increase customer satisfaction and retention.
- Increases profit margin and increase turnover.
- Improve employee motivation, awareness and morale.
- Reduce cost and increases productivity.
- Provides quick responses to change.
- Avoids over-production

Integrated Approach

Agile teams incorporate Kanban principles to maintain continuous improvement. For this the team's mindset has to be changed in same aspects during the development process. One such important factor is that the team should not just deliver code but should deliver tested and completed product. So, this type of transitions plays major role in the increased productivity. The lean way of development methodology started out to justify the agile way of development methodology but now it has come up in its own way.

Irrespective of the agile methodology being used, the teams / organizations follow the transition from agile to integrate agile and Kanban teams can take place by incorporating the following approaches (Jyothi & Rao 2011):

1. While the product is under development, the key factors that need to be taken into consideration are the concept of limiting work in progress (WIP) and the importance of flow through the system.

2. The product owner has to arrange all incoming work and help maintain a regular schedule of meetings to breakout user stories with the help of development team.

3. The stand-up meetings have to be conducted weekly for story planning and estimation. The development team and product owner should participate so the teams will work in synchronization.

4. The Kanban board can be used by the Kanban teams instead of agile story boards which maintain the user stories.

5. As the daily stand-up meetings and iterative planning meetings are the necessary activities to be followed by the iterative agile teams, the Kanban teams should also schedule 15 minutes for daily stand-up meeting (Jeffries *et al.* 2001).

6. Usually the iterative agile development teams are responsible for large-scale projects but the Kanban teams can be effective while working for small scale project along with bugs fixing.

2.3 Distributed Agile Development

Last decade witnessed agile software development become a common business practice for many software development organisations (Abrahamsson *et al.* 2002). At the same time, distributed software development has become a common business reality, where the level of distribution can range from the teams being located in the same city to those on different continents (Cheng *et al.* 2009). The forces of globalisation have increasingly moved software development from traditional, co-located model towards distributed, offshore models across geographical boundaries (Sangwan *et al.* 2007). These forces are driven by a number of factors such as lower development costs, improved network infrastructure, a move towards component-based architecture and increased time-to-market pressure (Carmel 1999). Such distributed project development is one of today's big challenges for organisations (Sangwan *et al.* 2007). But examples of agile software development methods in DD projects are rare. Some of the challenges involved in DD projects have been identified and solutions to them have been proposed (i.e. Paasivaara *et al.* 2009, Fitzgerald *et al.* 2006) but most of these solutions are based on stable market requirements. However, today's business environment faces dynamic requirements and uncertain implementation technologies. This has made software development organisations apply agile development approaches such as XP and Scrum (Cao *et al*. 2013, Passivaara & Lassenius 2006).

Paasivaara *et al.* (2009) discussed the advantages Scrum practices bring to distributed software development processes, some of which are summarised as follows:

- *Daily Scrum Stand up meetings:* brings transparency to a distributed project, reveals possible problems early on, creates contacts and encourages informal communication, especially between the sites. It is referred as the most useful Scrum practice for distributed projects

- *Weekly Scrum of Scrums:* distributes information between the teams, reveals possible problems early on, opens discussion channels and encourages informal communication between the teams
- *Sprints:* provides frequent monitoring opportunities between the sites
- *Sprint planning meeting:* give a possibility for the team members from all sites to participate, to seek clarification, to understand tasks and to commit to common goals. It also brings transparency to a distributed project
- *Backlogs*: All team members can access, pick up items and follow the progress

The above benefits indicate that not only are agile methodology practices applicable in DD concepts, but they are able to improve some of the DD challenges such as transparency and coordination (Dingsøyr & Moe 2013). In addition to these benefits, the literature identifies three strategies for distributed Scrum teams that are commonly used in practice (Sutherland *et al.* 2007):

- isolated Scrums where the teams are isolated across geographical distances
- distributed Scrum of Scrums where Scrum teams are isolated across geographical distances and integrated by a Scrum of Scrums that meets regularly across geographies
- totally integrated Scrums where Scrum teams are cross-functional and the teams distributed across various geographical distances

Best practice recommended by the Scrum Alliance is a Distributed Scrum of Scrums model (Sutherland *et al.* 2007). This model partitions work across cross-functional, isolated Scrum teams while eliminating most dependencies between the teams. Scrum teams are linked by a Scrum of Scrums where Scrum Masters (team leaders/project managers) meet regularly across locations. This encourages communication, cooperation, and cross-fertilisation.

2.3.1 Distributed Agile Development Challenges & Practices

The above section discussed the agile methodologies and the benefits of agile practices. However, agile methodology is not without problems and challenges. Various sources have noted challenges ranging from communication, collaboration, cultural differences to lack of team building and coordination challenges. These challenges are discussed below and summarised in Figure 2.2.

In the recent literature, it has been observed that there is a growing interest in applying agile practices to distributed software development projects (Hossain *et al.* 2009). However, such increasing movement toward distributed software development and the formation of distributed teams poses a potential dilemma for organisations who have adopted agile methods (Hildenbrand *et al.* 2008). This led scholars to combine distributed and agile software development into a common distributed agile development model (Mattsson *et al.* 2010, Passivaara & Lassenius 2006, Ramesh *et al.* 2006) for the purposes of examination. There is scant literature addressing the problems and challenges of applying agile methods in distributed software development projects. Recent studies encourage the researchers to further extend the quality and quantity of empirical studies to describe, evaluate, analyse and explore challenges and suggest solutions to these problems (Hossain *et al.* 2009, Ramesh *et al.* 2006, and Mattsson *et al.* 2010).

Agile and DD approaches differ significantly in their key principles. Agile methods and, in particular, agile practices have been designed for co-located software development (Cockburn 2002). However, DD project teams are located in different geographical distinct places. For example, while agile development approaches mainly rely on social processes to facilitate coordination, DD typically relies on formal mechanisms to manage and control the process (Ramesh *et al.* 2006). DD faces challenges in communication, lack of control and lack of trust: all areas that agile is typically strong in due to the close working relationships fostered through co-location and face-to-face communication (Ramesh *et al.* 2006). The following major conflicts for DD vs. agile are identified by Ramesh *et al.* (2006):

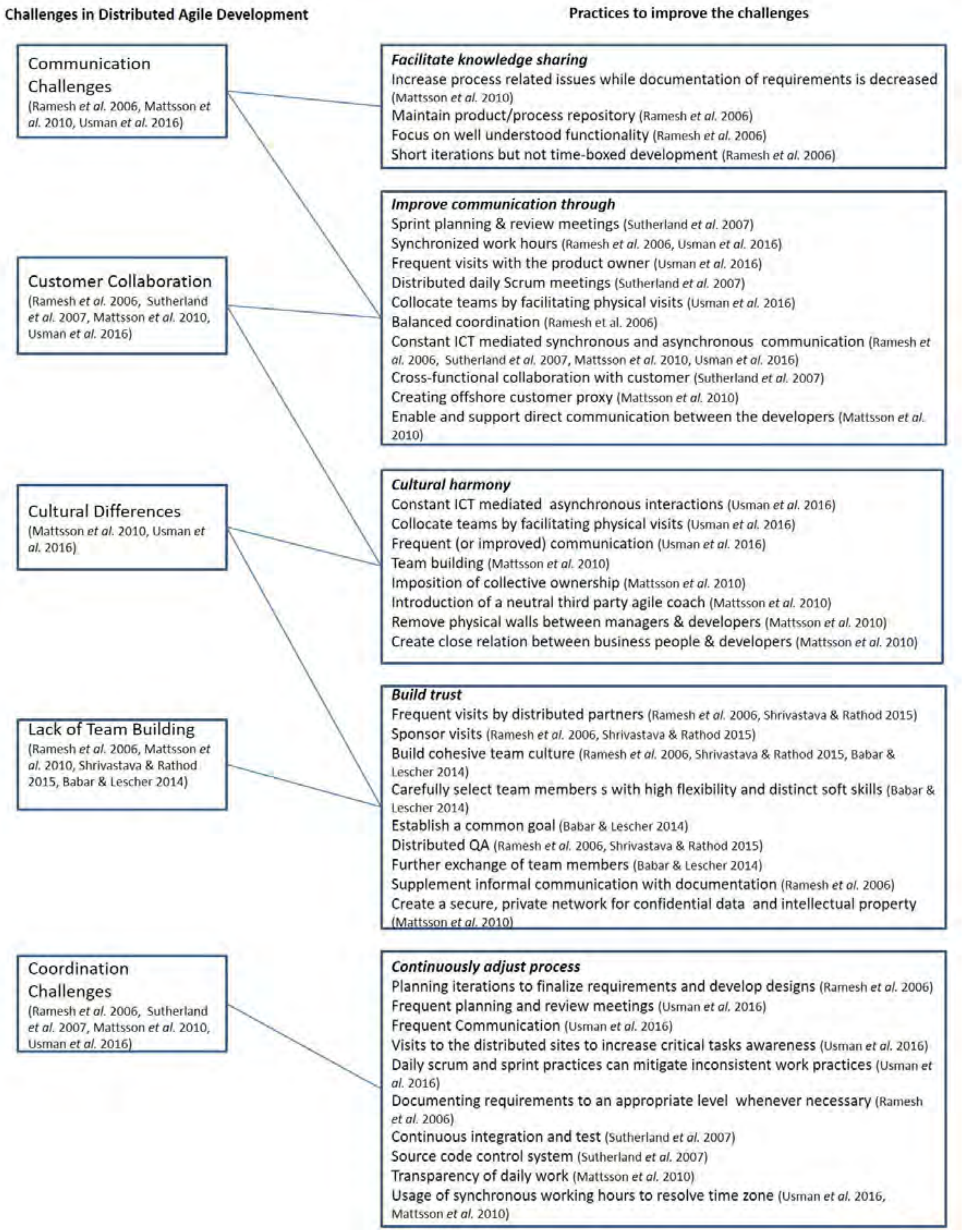
- *communication need vs. communication resistance:* distributed software development relies on formal communication but agile software development relies more on informal interactions

- *fixed vs. evolving quality requirements:* DD relies on fixed and planned quality requirements but agile software development relies on continuous negotiation between customer and developers to identify the requirements
- *people vs. process oriented control*: DD uses formal control processes but agile software development establishes control through informal (social) processes
- *formal vs. informal agreements*: DD contracts are explicitly defined but agile software development contracts are flexible and loosely defined
- *lack of team cohesion*: DD teams have less perception of belonging to the same team but agile software development teams have strong perception of belonging to the same team.

Key assumptions of agile development practices are challenged in distributed software development environments. Agile development methods emphasise self-organising, rich and continuous face to face communication, light documentation and close frequent collaboration among the project teams (Highsmith & Cockburn 2001). These characteristics are usually considered effective for co-located projects with a small team size (Abrahamsson *et al.* 2002). Indeed, within the context of distributed software development, the teams must rely on different technology tools to communicate and in some cases they never meet in person (Hildenbrand *et al.* 2008). The intense interaction between the developers and between the development team and customers that is typical of agile development projects suffers from the challenge of distance and may suffer further from cultural differences within the team (Ramesh *et al.* 2006).

This section discusses various challenges identified by scholars, particularly in regard to the application of Scrum to DD projects. The challenges and solutions proposed by scholars in literature are summarised in Figure 2.2.

Figure 2. 2 Summary of Challenges & Practices Suggested in Literature



Challenges in Distributed Agile Development

Communication Challenges (Ramesh *et al.* 2006, Mattsson *et al.* 2010, Usman *et al.* 2016)

Customer Collaboration (Ramesh *et al.* 2006, Sutherland *et al.* 2007, Mattsson *et al.* 2010, Usman *et al.* 2016)

Cultural Differences (Mattsson *et al.* 2010, Usman *et al.* 2016)

Lack of Team Building (Ramesh *et al.* 2006, Mattsson *et al.* 2010, Shrivastava & Rathod 2015, Babar & Lescher 2014)

Coordination Challenges (Ramesh *et al.* 2006, Sutherland *et al.* 2007, Mattsson *et al.* 2010, Usman *et al.* 2016)

Practices to improve the challenges

Facilitate knowledge sharing
Increase process related issues while documentation of requirements is decreased (Mattsson *et al.* 2010)
Maintain product/process repository (Ramesh *et al.* 2006)
Focus on well understood functionality (Ramesh *et al.* 2006)
Short iterations but not time-boxed development (Ramesh *et al.* 2006)

Improve communication through
Sprint planning & review meetings (Sutherland *et al.* 2007)
Synchronized work hours (Ramesh *et al.* 2006, Usman *et al.* 2016)
Frequent visits with the product owner (Usman *et al.* 2016)
Distributed daily Scrum meetings (Sutherland *et al.* 2007)
Collocate teams by facilitating physical visits (Usman *et al.* 2016)
Balanced coordination (Ramesh et al. 2006)
Constant ICT mediated synchronous and asynchronous communication (Ramesh *et al.* 2006, Sutherland *et al.* 2007, Mattsson *et al.* 2010, Usman *et al.* 2016)
Cross-functional collaboration with customer (Sutherland *et al.* 2007)
Creating offshore customer proxy (Mattsson *et al.* 2010)
Enable and support direct communication between the developers (Mattsson *et al.* 2010)

Cultural harmony
Constant ICT mediated asynchronous interactions (Usman *et al.* 2016)
Collocate teams by facilitating physical visits (Usman *et al.* 2016)
Frequent (or improved) communication (Usman *et al.* 2016)
Team building (Mattsson *et al.* 2010)
Imposition of collective ownership (Mattsson *et al.* 2010)
Introduction of a neutral third party agile coach (Mattsson *et al.* 2010)
Remove physical walls between managers & developers (Mattsson *et al.* 2010)
Create close relation between business people & developers (Mattsson *et al.* 2010)

Build trust
Frequent visits by distributed partners (Ramesh *et al.* 2006, Shrivastava & Rathod 2015)
Sponsor visits (Ramesh *et al.* 2006, Shrivastava & Rathod 2015)
Build cohesive team culture (Ramesh *et al.* 2006, Shrivastava & Rathod 2015, Babar & Lescher 2014)
Carefully select team members s with high flexibility and distinct soft skills (Babar & Lescher 2014)
Establish a common goal (Babar & Lescher 2014)
Distributed QA (Ramesh *et al.* 2006, Shrivastava & Rathod 2015)
Further exchange of team members (Babar & Lescher 2014)
Supplement informal communication with documentation (Ramesh *et al.* 2006)
Create a secure, private network for confidential data and intellectual property (Mattsson *et al.* 2010)

Continuously adjust process
Planning iterations to finalize requirements and develop designs (Ramesh *et al.* 2006)
Frequent planning and review meetings (Usman *et al.* 2016)
Frequent Communication (Usman *et al.* 2016)
Visits to the distributed sites to increase critical tasks awareness (Usman *et al.* 2016)
Daily scrum and sprint practices can mitigate inconsistent work practices (Usman *et al.* 2016)
Documenting requirements to an appropriate level whenever necessary (Ramesh *et al.* 2006)
Continuous integration and test (Sutherland *et al.* 2007)
Source code control system (Sutherland *et al.* 2007)
Transparency of daily work (Mattsson *et al.* 2010)
Usage of synchronous working hours to resolve time zone (Usman *et al.* 2016, Mattsson *et al.* 2010)

Adopted from Ramesh *et al.* (2006) Sutherland *et al.* (2007), Mattsson *et al.* (2010), Babar & Lescher 2014, Shirvastava & Rathod 2015, Usman *et al.* (2016).

To address the above challenges, researchers propose that firms have the capability to combine the flexibility offered by agile development approach with the planned elements of the traditional development approach (Ågerfalk *et al.* 2009, Ramesh *et al.* 2006). In other words, development organisations need to have the capability and flexibility to modify agile practices when they apply them to a distributed agile development project so that they are able to leverage the benefits of agile development approach while maintaining the values (Ågerfalk *et al.* 2009). Therefore, the main challenge of applying agile development methods in a distributed context would be to remain agile. For instance, frequent and continuous communication, short iterations, frequent builds and continuous integration can be strengths in a DD environment (Sutherland *et al.* 2007). Frequent and open communication between the project teams builds trust and the project teams can learn quickly about each other's cultures, thus reducing the probability of serious cultural conflicts (Passivaara & Lassenius 2006).

The benefits of short iterations are numerous. Here are two of the benefits discussed in the literature:

- Short iterations bring transparency of work progress to all stakeholders (Ramesh *et al.* 2006): While the customer is able to monitor real progress and frequently witness high quality work during the project, the developers are able to get immediate feedback on their work, which is motivating (Passivaara & Lassenius 2006). This by itself fosters trust and respect between project stakeholders that improves communication and makes collaboration much easier
- Frequent and continuous integration and testing of builds ensures that all the project teams have understood the requirements correctly: It is particularly more effective where the project teams have not worked together before and have different cultures (Abrahamsson *et al.* 2002). Continuous integration and testing also provides developers with fast feedback therefore, any misunderstanding would be visible early with less damaging consequences (Sutherland *et al.* 2007)

The study conducted by Sutherland *et al.* (2007) suggested various practices for distributed Scrum projects to overcome the DD challenges. These practices were applied to a project where an excellent implementation of Scrum, along with one global build repository, one tracking and reporting tool and daily meetings across geographies were all implemented:

- daily Scrum team meetings of all developers from multiple sites through video conferencing at a common time for every site

- daily meetings of Product Owner team
- hourly automated builds from one central repository
- no distinction between developers at different sites on the same team
- rare integration of XP practices like pair programming with Scrum

Some scaling mechanisms, such as daily Scrum of Scrums that coordinates the work of Scrum teams, enabled the application of large Scrum projects involving multiple Scrum teams possible (Schwaber 2004). In these kinds of projects, synchronised builds and alternative communication methods (such as instant messaging) need to be deployed (Schwaber 2004).

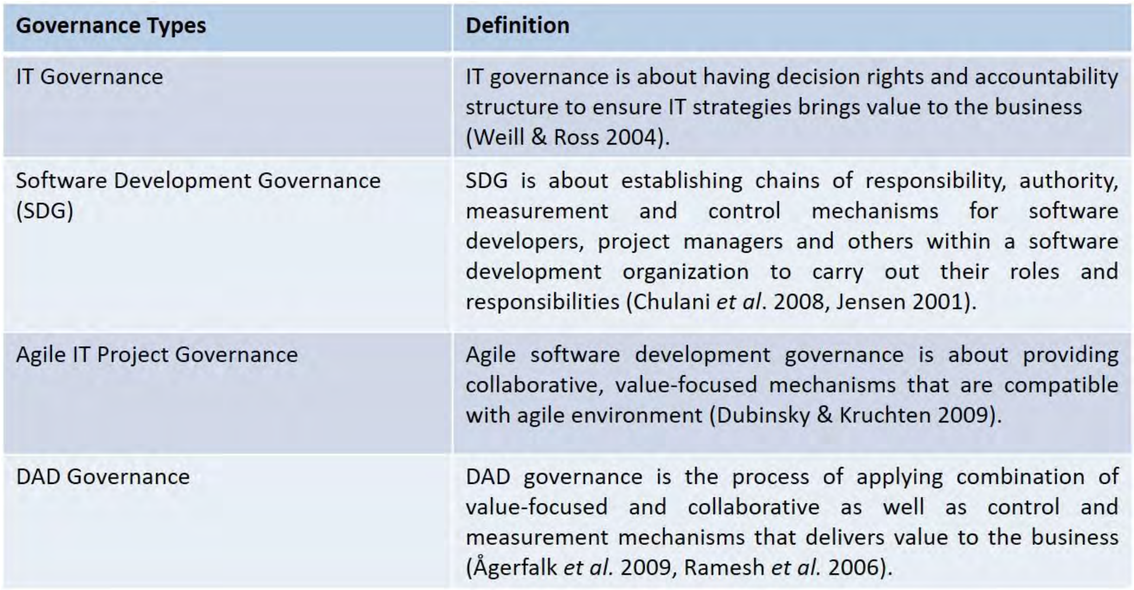
Mattsson *et al.* (2010) studied 12 empirical case studies that mainly used XP or Scrum methods. In this study I identified distributed agile development problems and categorised them into six classes: culture, time zone, communication, customer collaboration, trust, training and technical issues. Similar findings have been revealed in the study conducted by Hossein *et al.* (2009).

Based on the limited number of studies in the literature it seems that it is possible to apply an agile methodology to distributed projects when there is an excellent Scrum implementation and frequent communication is arranged efficiently. However, the suggested solutions to distributed challenges are scarce and need a more holistic view of controlling and governing them. As distributed agile development becomes mainstream and becomes institutionalised, new issues will arise, such as questions of governance in distributed agile development projects. Taking the above challenges into consideration, one can conjecture that the more distributed the software development project is over time and space then the less compatible distributed agile development is with agile practices. Therefore, an appropriate governance framework is required to reduce DD challenges. Rather than add to the loss of agility by imposing formal governance mechanisms on to distributed agile development, the current study proposes that social governance can be used to preserve and maintain agility in distributed agile development. In the next section, governance in IT and software development context is discussed.

2.4 Governance Literature

In the corporate sector, the term corporate governance refers to the way in which corporations and businesses administer and control a corporation to the satisfaction of major stakeholders and regulators (Bannerman 2009). Borrowing from the discipline of corporate governance, the term IT governance has been discussed in the last decade. At the IT level, governance research has tended to focus on forms of governance such as the structuring of IT activities and locus of IT decision-making within organisations (Bannerman 2009).

Table 2. 3 Governance Types



Governance Types	Definition
IT Governance	IT governance is about having decision rights and accountability structure to ensure IT strategies brings value to the business (Weill & Ross 2004).
Software Development Governance (SDG)	SDG is about establishing chains of responsibility, authority, measurement and control mechanisms for software developers, project managers and others within a software development organization to carry out their roles and responsibilities (Chulani *et al*. 2008, Jensen 2001).
Agile IT Project Governance	Agile software development governance is about providing collaborative, value-focused mechanisms that are compatible with agile environment (Dubinsky & Kruchten 2009).
DAD Governance	DAD governance is the process of applying combination of value-focused and collaborative as well as control and measurement mechanisms that delivers value to the business (Ågerfalk *et al.* 2009, Ramesh *et al.* 2006).

In order to outline the requirements of governance in distributed agile development projects, I first considered the principles of IT governance and then discussed how these principles were applied to software development as software development governance (SDG). Next, I explain governance of agile IT projects and raise the question of what governance mechanisms are appropriate. Finally, I propose the appropriate governance mechanisms for distributed agile development projects. Table 2.3 summarises different governance types discussed in this study.

2.4.1 IT Governance

Complexity in information technology architectures and infrastructures, and an increasing need for firms to align their IT organisations with their business to secure value generation processes, calls for an increasing awareness and understanding of IT Governance (Pooley *et al.* 2013). Although various definitions of IT governance have been proposed, a widely accepted view in the literature is that IT governance is the top management concern in the allocation of decision making rights, accountability and the establishment of a formal reporting structure (Ribbers *et al.* 2002). In other words, IT governance is about directing and controlling the use of IT within an organisation to ensure that IT investments create business value for the organisation - which ultimately contributes to organisational viability and desired outcomes (Ribbers *et al.* 2002, Weill & Ross 2004).

There is increasing evidence that effective governance delivers value (Massa & Tucci 2013). The literature on IT governance suggests that organisations with effective IT governance structures tend to have better performance, because appropriate IT governance structures promote desirable IT behaviours by directing, controlling, and coordinating IT activities (Pye & Warren 2006, Khther & Othman 2013). Effective governance is identified as the single most important predictor of the value that an organisation generates from its IT activities (Weill & Ross 2004, Raj *et al.* 2013). In addition, it has been found that organisations with effective IT governance programs gained twenty percent higher profit margins than those organisations with poor quality governance programs that had similar strategic goals (Pye & Warren 2006). Firms achieving above average returns from IT investments must be making consistently better IT decisions, and effective governance is one of the ways that these organisations can achieve such returns (Weill and Woodham 2002).

One of the first academic studies to provide a more detailed investigation into IT governance adopted by large, complex organisations was conducted by Brown (1999). In this study, the formal, top-down mechanisms used to coordinate IT activities across geographical boundaries between a firm's corporate headquarters and its component divisions were investigated (Brown 1999). The study argued that coordination and collaboration mechanisms were required to enable organisations to respond effectively to the challenges posed by an increasingly complex and uncertain environment. Drawing on Organisation Theory research, the study provided a comprehensive list of mechanisms that may be used to coordinate activities within complex organisations (Brown 1999). This study suggests that coordination mechanisms are essential for responding effectively to DD challenges. However, the study does not cover governance mechanisms for distributed agile software development projects.

A subsequent study that also incorporated coordination mechanisms into its conceptualisation of IT governance was conducted by Peterson (2001). The study argued that the allocation of specific decision rights to different organisational units required the coordination of those decisions across those organisational units. The study proposed three categories of integration mechanisms: structural, functional and social. A key finding of the study was that hybrid structures lacking in effective coordination mechanisms were associated with poor IT performance. Therefore, the literature studies suggest coordination mechanisms could possibly be an important element of governance in order to resolve DD challenges.

The comprehensive study performed by Weill and Ross (2004) identified the various structural configurations and governance mechanisms used by contemporary organisations to enact their IT governance arrangements. This study investigated the structuring of IT activities and IT decision-making within organisations by characterising IT governance as addressing these three things: what IT decisions must be made, who has the decision rights and how these decisions should be made. The study further argues that when firms work hard to generate value from their IT investment, managers become increasingly aware that IT related decisions and behaviours must be aligned with organisational performance goals. Many individual managers throughout organisations make daily decisions influencing the value received from IT. The study also provided practitioners with broad guidelines for designing their IT governance systems.

Overall, the literature on IT governance emphasises the benefits and value that firms can gain by applying effective IT governance principles to align their IT and business goals and activities (Weill & Ross 2004). Effective IT governance must address the following questions (Weill & Ross 2004):

1. What decisions must be made to ensure effective management and use of IT?
2. Who should make these decisions?
3. How are these decisions made and monitored?

The key principles of effective IT governance are summarised below:

- Transparency (Weill & Ross 2004); make the governance mechanism transparent to all stakeholders

- Actively design governance (Weill & Ross 2004, Weill & Woodham 2002); IT governance must be designed with the desirable behaviours in mind
- Know when to redesign governance (Weill & Ross 2004, Weill & Woodham 2002); redesign infrequently and only when needed
- Educate about governance (Weill & Woodham 2002); education to help stakeholders understand and use the governance mechanisms is critical
- Good governance requires choices (Weill & Woodham 2002); effective IT governance structures are simple and attempt to optimise a small number of performance goals and metrics
- Handle exceptions (Weill & Woodham 2002); effective IT governance needs to provide a process for handling exceptions for new opportunities

While most of these principles have focussed on the organisational level, SDG, a relatively new term in the software development literature, extends governance principles from the organisation / IT governance level to the software development level (Yaeli & Klinger 2009). IT governance is concerned with various IT aspects of a firm, within which the governance of software development is an important area.

2.4.2 Software Development Governance (SDG)

The previous section discussed IT governance and the key principles and advantages of effective IT governance. This section extends the IT governance concept and considers how governance might apply to the software development domain. The following discusses the objectives of SDG, major aspects of SDG and explains different approaches of SDG.

An important IT activity that can benefit from the application of governance principles is software development (Jensen 2001). That is how the concept of IT governance has been extended to more operational levels in organisations as software development governance (SDG) (Yaeli& Klinger 2009). SDG is concerned with the management of software and system development projects in development organisations (Bannerman 2009).

SDG helps to align the software development activity with business objectives and the strategic requirements of the organisation, as defined by enterprise governance, to create business value for the organisation (Yaeli& Klinger 2009). Thus, effective SDG enables the projects to achieve satisfactory outcomes and create business value (Chulani *et al.* 2008). It also helps mitigate risks and improves the team effectiveness by enabling good communication and effective measurement and control (Chulani *et al.* 2008).

The major purpose of SDG is to establish how the organisation's software development capability is sustained, in terms of structures, processes and relational mechanisms, to meet its engineering and business needs (Bannerman 2009). The goals of SDG are to increase predictability, safeguard value realisations, manage risk and change and provide clarity and accountability (Tarr *et al.* 2008) in software development projects.

SDG defines organisational structures through establishing chains of responsibility, authority and communication to empower people within a software development organisation (Chulani *et al.* 2008). Further, SDG achieves its goals and objectives by establishing measurement and control mechanisms to enable software developers, project managers and others within a software development organisation to carry out their roles and responsibilities (Jensen 2001).

The major aspects of governance in software development organisations are (Dubinsky *et al.* 2008):

- setting goals and assigning responsibilities and decision rights to make individuals accountable for their actions to meet the goals
- providing and executing guidelines and control mechanisms to constrain performance dynamics and opportunistic behaviours

One of the main aspects of SDG is having control mechanisms to safeguard performance and actions. However, to achieve these goals, SDG has to address a number of challenges, such as managing the value through aligning business, balancing risk and providing clarity and accountability (Biffl *et al.* 2005). For example, Boehm (2008) presents a value-based process framework and life cycle process model for achieving effective system and software development governance, called the Incremental Commitment Model (ICM). This process framework is based on underlying principles including:

- commitment and accountability
- success-critical stakeholder satisfaction

- incremental growth of system definition and stakeholder commitment
- iterative system definition and development cycles
- risk-based activity levels and anchor point commitment milestones

The study discusses ICM in project governance and extends to enterprise governance in order to justify resource commitments and to monitor progress with respect to business objectives.

Another example is the study conducted by Yaeli and Klinger (2009) that presents the concept of responsibility assignment to the software development process. Responsibility assignment is mostly concerned with who should make these decisions and how these decisions will be made and monitored. There are various ways of representing role assignments including Yaeli and Klinger's (2009) popular RACI matrix: (R)esponsible, (A)ccountable, (C)onsulted, and (I)nformed. RACI matrices are used to document the role and responsibility assignments for the teams and stakeholders for performing activities. The study discussed common representations of responsibility assignments, their relationship to the operational model of software engineering including an artefact lifecycle based approach, and the governance points through which governance solutions join with the operational models to realise governance enactment.

Taking the above studies into consideration, along with the diversity of software development organisations in terms of their size, activities and strategic goals (Lindvall & Rus 2000), the conclusion can be drawn that there is no single solution to governance challenges that fits all organisational circumstances. For example, the research conducted by Bannerman (2009) suggests that governance of development tools and operating systems software needs to be highly centralised to ensure coordination and integration of products with target platforms, component specifications and standards.

On the other hand, Bannerman also suggests that the governance of customised applications software is better decentralised, in order to facilitate closer interaction with customers. The study concludes that, to have an effective and value added governance, an organisation requires the governance mechanism to fit the organisation's business model, environmental context, stage of growth and size. Therefore, different firms require governance structures that are designed and developed with the specific circumstances of the organiwation in mind.

In the software development literature, some studies (Erbas & Erbas 2009, Tarr *et al.* 2008) investigated governance through different software development activities. Two key approaches to governance were identified, being top-down or planned (formal) governance and bottom-up or adaptive (social) governance (Tarr *et al.* 2008). In top-down software process development, the focus is on decomposing the system into smaller modules and specifying the interface between modules through contracts, then development of each module is done independently. However, in the bottom-up development process, instead of following up-front planned processes, as soon as the first module is identified, the coding and testing can commence, which empowers the development team with regards to their role, efforts required and techniques they would like to use; therefore the cooperation between parties is very important and can improve the integration phase in later stages (Erbas & Erbas 2009).

The bottom-up approach appears compatible with agile development methodology because it empowers the development team and gives individuals the decision making rights for effort estimation, assigning resources and choosing the techniques for development, which all require a close coordination among the teams (Tarr *et al.* 2008). Therefore, coordination among the teams in agile software development is an important element. Reports indicate a rapid and steady adoption of agile software development projects (Schwaber *et al.* 2007, Svensson & Höst 2005, Manhart and Schneider 2004) which indicates there is a need for governance approaches to govern these projects.

The bottom-up development processes are more cost effective when software development organisations are dealing with more application specificity and uncertainty (Keng 2011). The cost effectiveness of top-down development processes increase as the application specificity and uncertainty decreases (Erbas & Erbas 2009). Therefore, the study suggests an effective software governance program should provide support for both top-down and bottom-up development processes.

This section discussed two different approaches of SDG in development firms; top-down and bottom-up. Characteristics of agile software development processes require a different view and goal of governance to be followed because there is no planned or designed process to ensure that they exhibit certain properties, deliver certain value, or understand and manage the risk they pose. The next section discusses in detail the agile IT project governance characteristics and models suggested in the literature.

2.4.3 Agile IT Project Governance

As has been in discussed, the governance structure in IT governance and SDG is formal. However, although formal governance could be a useful approach, by imposing top-down command and control structures it may work against the agile development 'ethos', in which autonomy and self-organisation are prized (Ambler 2009). Therefore, an effective governance approach for agile IT projects needs to be based on collaboration, motivation and value-focussed strategies, with due regard for social governance mechanisms (Dubinsky & Kruchten 2009). Agile governance involves social mechanisms with light weight, collaborative, coordination oriented framework that empowers agile IT project teams to create business value by strategic alignment of 'business-agile' goals (Cheng *et al.* 2009).

Qumer (2007) proposes an agile governance model that emphasises the issue of maximising the business value (return on investment). This is done through strategic alignment of business and agile goals, by focussing on decision making and accountability and by assessment frameworks for performance and risk management where every individual is accountable for their roles and responsibilities. This view of governance in agile software development is particularly compatible with the principles and characteristics of agile project teams. Qumer (2007) argues that an effective agile governance approach will facilitate the achievement of desired discipline, business value, improved performance, as well as control in large agile software development environments by aligning business goals and agile software development goals. However, the study does not cover detailed agile governance mechanisms and framework to support large / distributed agile environments.

Following Qumer (2007), Ambler (2009) suggests that the self-organising characteristic of an agile team makes individuals within the team accountable for the project's scope despite being faced with frequent changes in requirements. Agile teams are accountable for budget because each iteration the agile project teams go through requires estimation of effort and required resources. Agile teams are also accountable for project schedules because in each iteration agile project teams need to decide the schedule for that iteration.

Further, Talby and Dubinsky (2009) state that governance mechanisms which enable managers to monitor projects effectively are the key success factors to increasing the business value. The study suggests that, in order to improve the quality of software products, governance mechanisms need to be performed iteratively. Implementing governance to match the short development iterations provides the means for a continuous reflection of the key issues in the development project. The study found that governance is tight and effective when performed at development iteration and governance events can shape the responsibilities of role holders and hence improve the role scheme in use. The study analyses agile governance at the level of single iterations only and not at release and full project level. The outcome of Talby and Dubinsky's (2009) study suggests that governance iterations can be combined with agile development iterations, resulting in a governance mechanism that identifies and resolves issues in an effective and timely manner. The study also does not propose any social mechanisms for governance of distributed agile projects.

Other studies draw more on agile characteristics. For example Dubinsky and Kruchten's (2009) study of the governance of agile teams suggests that the greater existence of visibility and opportunities to guide and steer the stakeholders in agile projects makes agile project teams easier to govern as compared with traditional teams. They argue that greater visibility results in the accomplishments of the project team being made explicit and therefore makes it easier for stakeholders to exercise governance of agile project teams.

While some (Weill & Ross 2004, Pye & Warren 2006, Yaeli & Klinger 2009) argue for a further cascade of IT governance principles from software development governance to agile governance, it seems that others (Dubinsky & Kruchten 2009, Cheng *et al.* 2009) conclude that agile development naturally lends itself to more informal, social modes of governance due to the visibility of the process, self-organising teams, and frequent delivery. A concern that does not appear to have been raised in the literature is that the adoption of formal governance mechanisms, i.e. the structures, processes and controls as the only governance framework in agile software development projects, may work against the very essence of agility. Next, distributed agile development governance characteristics are discussed in detail.

2.4.4 Distributed Agile Development Governance

As discussed in section 2.4.3, agile projects require social governance mechanisms that are compatible with the flexibility and empowerment that exists in an agile development environment. Complexity in distributed information technology architectures and infrastructures, and an increasing need for executives to verify and secure the value generation processes in organisations calls for an increasing awareness and understanding of IT governance in general and agile governance in particular (Ågerfalk *et al.* 2009). Consequently, as organisations expand their operations and their development projects become more distributed, the need for effective governance becomes more critical and governance approach is even more complex when it is a distributed agile development project (Ramesh *et al.* 2006). In light of the above discussion, the study concludes that literature and case studies are scarce in the investigation of the effective governance of agile software projects in distributed environments and precedent researches do not propose a holistic approach for governance of distributed agile development projects.

Based on the above discussion, the current study proposed the following research question:

RQ1) *How are social governance mechanisms applied in distributed agile development projects?*

Recently, the Carnegie Mellon Software Engineering Institute (Anderson & Carney 2009) proposed a Distributed Project Governance Assessment process, taking external dependencies of a distributed software project into consideration to manage the project risks involved. The study argues that in today's world of network and inter-organisational relationships, external dependencies can threaten development projects and therefore they need to have an effective governance mechanism to control these risks.

However, there is still scant literature to address questions such as:

1. *How can distributed agile development projects be governed*?
2. *What is an effective governance approach for distributed agile development projects?*
3. *What governance mechanisms do distributed agile development projects use and how are these mechanisms applied to distributed agile development projects in practice?*

This research aimed to address these gaps in literature by developing an appropriate governance approach for distributed agile development projects.

To address the DD challenges such as coordination, communication, lack of control, customer collaboration, cultural differences, lack of trust and technology (Ramesh *et al.* 2006, Sutherland *et al.* 2007 and Mattsson *et al.* 2010), researchers propose that firms must combine the flexibility offered by the growing agile development approach with the rigidity offered by the traditional plan-based approach, thus this study sought to combine those two into a common distributed agile development model (Ågerfalk *et al.* 2009; Ramesh *et al.* 2006). Providing such an agile-rigid environment can help the development organisations mitigate different risks inherent in distributed software development projects (Yadav *et al.* 2007). Therefore, a theory is required to bind these informal governance mechanisms together. Next the current study discusses network governance theory and presents a framework for distributed agile development projects.

2.5 A Framework for Distributed Agile Development Project Governance

To develop an agile-rigid model of distributed agile development governance, this section theorises social governance through NG theory and studies the application of social mechanisms offered by NG theory, then proposes a research model.

2.5.1 Network Governance Theory

As discussed previously (see sections 2.4.3 and 2.4.4), an effective governance approach for agile development projects needs to be compatible with collaborative and value-focussed strategies that exist in agile development context. This suggests that social governance mechanisms are a suitable candidate for such a purpose (Dubinsky & Kruchten 2009, Cheng *et al.* 2009). Therefore, to explore social governance, the current study drew on NG theory, which is “characterized by informal social systems rather than bureaucratic structures within firms, and formal contractual relationships between them” (Jones *et al.* 1997, p. 911).

NG theory is an integration of TCE and Social Network theories (Jones *et al.* 1997). The NG theory says that the network (social) form of governance is a response to exchange conditions of asset specificity, demand uncertainty, task complexity and frequency (Williamson 1994). These exchange conditions drive organisations toward structurally embedding their transactions. Structural embeddedness enables the use of social mechanisms for coordinating and safeguarding exchanges between the project teams (Granovetter 1992, Jones *et al.* 1997). NG theory provides a theoretical framework for examining how the efforts of distributed agile development project teams can be coordinated and safeguarded effectively.

Previously (see Sections 2.4.1 and 2.4.2) the current study explained that the literature discussed coordination and safeguarding as important elements of governance to reduce DD challenges (Brown 1999, Peterson 2001, Cheng *et al.* 2009, Tarr *et al.* 2008 and Bannerman 2009). Indeed, the literature suggests that coordination oriented governance frameworks empower agile project teams to create business value (Cheng *et al.* 2009). In addition, coordination is identified as one of the main practices in the literature that reduces DD challenges (Ramesh *et al.* 2006, Sutherland *et al.* 2007 and Mattsson *et al.* 2010).

Jones *et al.* (1997) have produced a model of network governance that provides a structure for studying informal governance mechanisms in self-organising organisations. Social mechanisms are used in the model to coordinate and safeguard the exchanges in the network in an efficient and effective manner. The model has not been widely taken up in information systems discipline, although notable exceptions are Feller *et al.* (2004), where it was used to study the role of social mechanisms in OSS settings. It was found that social mechanisms enable better coordination and ultimately enable a project to be more successful (Sagers 2004).

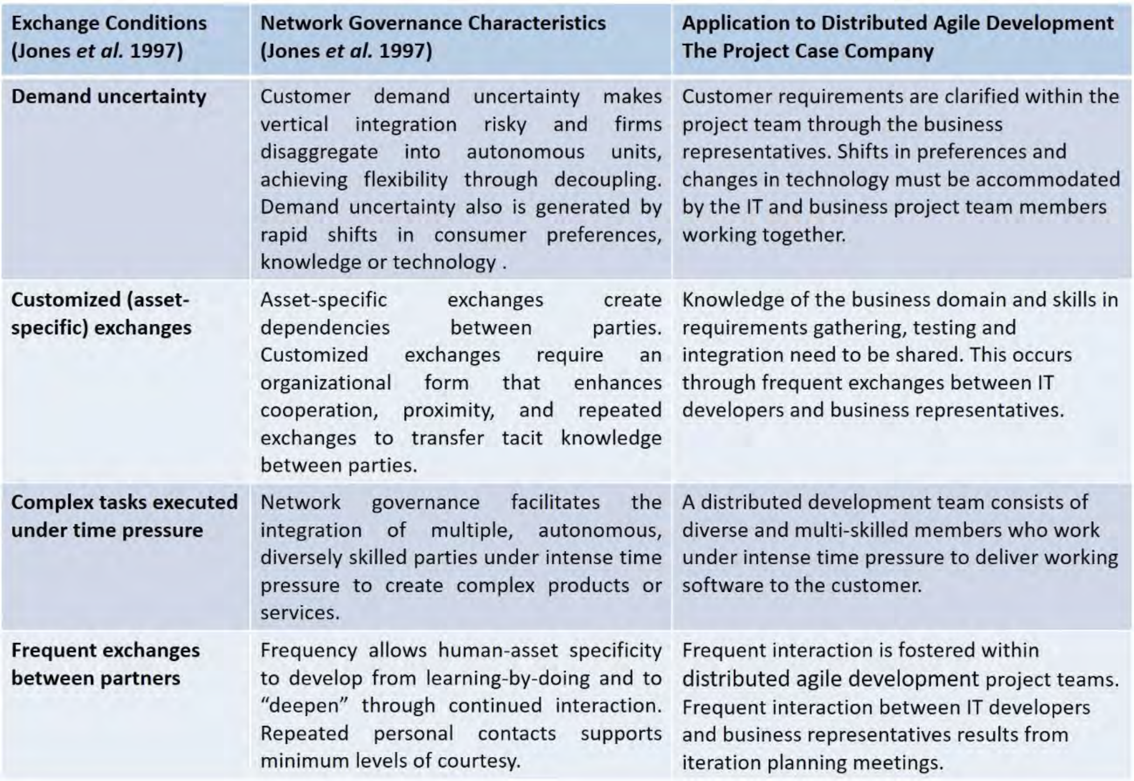
I adopted a governance framework for distributed agile development projects based on the Jones *et al.* (1997) NG model. I proposed that distributed agile development project teams can be conceptualised as using social mechanisms to coordinate activities and safeguard exchanges. This is because the essential characteristics of social structures and individual participation in creating a software product in a complex and uncertain environment are present in distributed agile development project teams. Consequently NG is an appropriate way of structuring our thinking about the informal and bottom-up aspects of distributed agile development project governance.

To demonstrate the appropriateness of the NG model in more detail, next section explains four conditions required for an effective NG to emerge and how these conditions are applicable to distributed agile development projects. Further, the characteristics of the Jones *et al.* (1997) model of social mechanisms that coordinate activities and safeguard exchanges are discussed along with their applicability to agile project teams. Given that an agile project team comprises IT developers and business representatives, where the IT personnel represent the interests of the technology and the business representatives look after the interests of the business (Schwaber & Beedle 2002), these representatives need to work together as a single entity, or network, to achieve business value for the customer (Baskerville *et al.* 2001). The customer is the business owner of the project and is accountable for the value created by the project. In some projects the business representative and the customer may be the same person, but they have separate and distinct roles (Schwaber & Beedle 2002).

2.5.2 Exchange Conditions

There are four preconditions required for NG to emerge and be effective: 1) demand uncertainty; 2) customised (asset-specific) exchanges; 3) complex tasks executed under time pressure; and 4) frequent exchanges between the project teams (Jones *et al.* 1997). Under such conditions, networks develop structural embeddedness (the extent to which a "dyad's mutual contacts are connected to one another" (Granovetter 1992, p. 35), creating both direct and indirect ties between parties, and increasing the visibility of the parties' activities. The presence of high levels of structural embeddedness and visibility of activities enables the use of various social mechanisms that resolve exchange problems by coordinating and safeguarding exchanges within networks (Jones *et al.* 1997).

Table 2. 4 Exchange Conditions of Effective Governance for Distributed Agile Development Project Teams



Exchange Conditions (Jones *et al.* 1997)	Network Governance Characteristics (Jones *et al.* 1997)	Application to Distributed Agile Development The Project Case Company
Demand uncertainty	Customer demand uncertainty makes vertical integration risky and firms disaggregate into autonomous units, achieving flexibility through decoupling. Demand uncertainty also is generated by rapid shifts in consumer preferences, knowledge or technology .	Customer requirements are clarified within the project team through the business representatives. Shifts in preferences and changes in technology must be accommodated by the IT and business project team members working together.
Customized (asset-specific) exchanges	Asset-specific exchanges create dependencies between parties. Customized exchanges require an organizational form that enhances cooperation, proximity, and repeated exchanges to transfer tacit knowledge between parties.	Knowledge of the business domain and skills in requirements gathering, testing and integration need to be shared. This occurs through frequent exchanges between IT developers and business representatives.
Complex tasks executed under time pressure	Network governance facilitates the integration of multiple, autonomous, diversely skilled parties under intense time pressure to create complex products or services.	A distributed development team consists of diverse and multi-skilled members who work under intense time pressure to deliver working software to the customer.
Frequent exchanges between partners	Frequency allows human-asset specificity to develop from learning-by-doing and to "deepen" through continued interaction. Repeated personal contacts supports minimum levels of courtesy.	Frequent interaction is fostered within distributed agile development project teams. Frequent interaction between IT developers and business representatives results from iteration planning meetings.

In Table 2.4, the characteristics of the four NG theory preconditions (referred to as exchange conditions) for an effective governance are described and then the study presents the application of these preconditions applied to the agile project case in the current study. The study argues that distributed agile development project teams, through their mode of organisation and values, should, if applied faithfully, lead to the structural embeddedness that is necessary to establish a NG social mechanism.

2.5.3 Social Governance Mechanisms

In an uncertain and competitive environment with high adoption, high coordination and high safeguarding of exchanges, the project teams are required to coordinate in order to develop complex products and services (Grover *et al.* 2002). Such coordination relies heavily on social components such as trust and relationship-specific assets (Kollock 1994). In the current study, social governance mechanisms refer to bottom-up governance mechanisms that respect empowerment, democracy, coordination and self-organising in development project teams. Once these exchange conditions have been met, Jones *et al.* (1997) identified four social mechanisms that influence exchange behaviour: (1) macroculture; (2) restricted access; (3) collective sanctions; and (4) reputation (Table 2.6). These mechanisms facilitate the adaptation, coordination and safeguarding of exchanges among the project teams.

Macroculture is concerned with shared beliefs and the assumptions and values comprising occupational or professional knowledge. Shared values and norms, including technical and professional knowledge, establish behavioural patterns among partners. Coordination costs are reduced by the convergence of values and beliefs through socialisation, establishment of common language and shared tacit rules about behaviour because the behavioural ground rules do not need to be re-established every time (Jones *et al.* 1997). These shared tacit rules are critical for effective exchange among different partners (Abrahamson & Fombrun 1992, Jones *et al.* 1997).

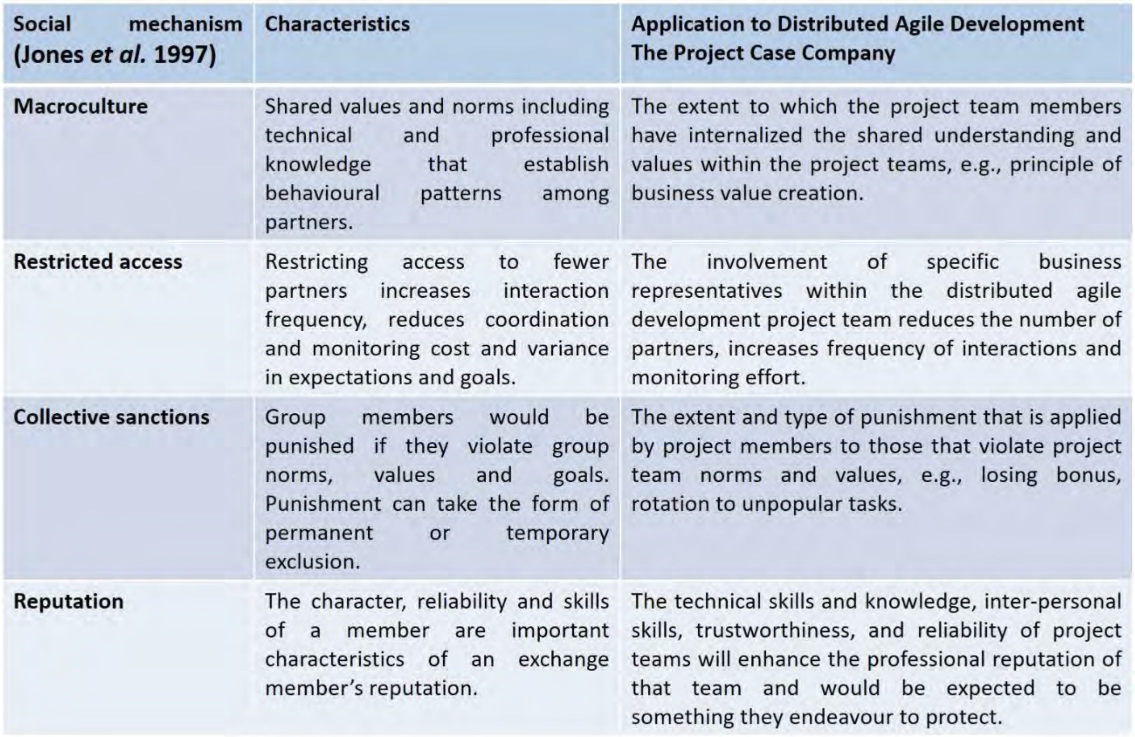
Restricted access is concerned with reducing the number of strategic partners in the network (Jones *et al.* 1997). Avoiding particular partners reduces potential performance risks (Das & Teng, 1996) while enabling more frequent interactions with the remaining partners. Fewer partners reduce coordination costs and the amount of monitoring needed, while enhancing the monitoring of others, and variances in the expectations, skills and goals the partners bring into an exchange (Jones *et al.* 1997). This also lessens the chance of opportunistic behaviours by the project teams. Restricted access in distributed agile development project teams means that the amount of monitoring of teams can be decreased. This is particularly important for distributed agile development projects where not all the teams are co-located. Restricting access also leads to an increase in commitment of the teams to the development project as stronger bonds are created between the teams.

Collective sanctions refer to punishments imposed by group members on other group members for violating norms, values or goals (Jones *et al.* 1997). Punishment can take the form of permanent or temporary exclusion from the network and could be for things such as sabotage or spreading gossip and rumours (Williamson 1994). Collective sanctions define and reinforce acceptable behavioural patterns by demonstrating the consequence of norm violation (Jones *et al.* 1997). They also increase the cost of misfeasance while decreasing the costs of monitoring.

Reputation is a perception or opinion someone holds about another person in regards to the person’s character, knowledge, skills and reliability (Jones *et al.* 1997). In highly uncertain environments, people are increasingly concerned about their own and others' reputations (Kollock 1994). Reputation is critical for a person to be repeatedly invited to participate in exchanges (Williamson 1994). At the same time, one's reputation is impaired if they are recommended and fail to meet the expectations of the network participants. Reputation safeguards exchanges by spreading information about behaviour among parties.

In distributed agile development project teams, restricting access to only business representatives that have been agreed as points of contact will result in more frequent interaction between agile IT developers and these business representatives. This, in turn, reduces coordination costs and safeguarding as relationships are established between parties and, in time, trust also develops between the parties (Jones *et al. 1997*). Safeguarding an interaction suggests that there is something to lose (reputation) and a price to pay if norms are not adhered to (collective sanctions) (Williamson 1994). These apply to both IT developers and business representatives as there is a chance of opportunistic behaviour and not being faithful from both sides. Table 2.5 summarises the characteristics of social mechanisms and their application to the agile project case in the current study.

Table 2. 5 Social Mechanisms for Distributed Agile Development Project Governance



Social mechanism **(Jones *et al.* 1997)**	Characteristics	Application to Distributed Agile Development The Project Case Company
Macroculture	Shared values and norms including technical and professional knowledge that establish behavioural patterns among partners.	The extent to which the project team members have internalized the shared understanding and values within the project teams, e.g., principle of business value creation.
Restricted access	Restricting access to fewer partners increases interaction frequency, reduces coordination and monitoring cost and variance in expectations and goals.	The involvement of specific business representatives within the distributed agile development project team reduces the number of partners, increases frequency of interactions and monitoring effort.
Collective sanctions	Group members would be punished if they violate group norms, values and goals. Punishment can take the form of permanent or temporary exclusion.	The extent and type of punishment that is applied by project members to those that violate project team norms and values, e.g., losing bonus, rotation to unpopular tasks.
Reputation	The character, reliability and skills of a member are important characteristics of an exchange member's reputation.	The technical skills and knowledge, inter-personal skills, trustworthiness, and reliability of project teams will enhance the professional reputation of that team and would be expected to be something they endeavour to protect.

Based on the above discussion, the current study then proposed the following research questions:

RQ2) *How do social governance mechanisms coordinate exchanges among distributed agile development project teams?*

RQ3) *How do social governance mechanisms safeguard exchanges among distributed agile development project teams?*

2.5.4 Effective Governance through Coordination and Safeguarding

According to the TCE framework, governance mechanisms are used to coordinate activities and safeguard exchanges against opportunistic behaviours (Granovetter 1992). In a project network structure, the project teams are connected to each other, they coordinate their activities, adapt to environmental influences and protect their interactions against any intrusion, freeloading and losses due to opportunism (Williamson 1994). An effective network is one which provides an easy and safe way for their members to interact and work together (Jones *et al.* 1997).

Grover *et al.* (2002) argue that social governance mechanisms extend the motivations of trust and safeguarding of transactions from economic motivations to appreciation of each party's transaction reputation with long term perspective for future exchange. Distributed agile development projects can also benefit from such shifts in governance approach to face challenges like lack of communication, lack of control and lack of trust (Qumer 2007). More emphasis on relationship investment among the project teams can act as a base to build trust and share knowledge in distributed environments (Grover *et al.* 2002). Such trusted relationships reduce monitoring costs and provide the opportunity for distributed agile development project teams to generate more value-added activities.

Two exchange mechanisms, coordination and safeguarding, facilitate an environment where cooperative behaviours among teams can evolve (Jones *et al.* 1997). Following Feller *et al.* (2008), the current study defines coordination as "the ease with which interactions between teams are conducted" and safeguarding as "the degree to which interactions between teams are protected" (Feller *et al.* 2008, p. 478). Governance structures serve to safeguard exchanges against opportunistic behaviours where individuals are given to maximising self-interest and transacting with a tendency to achieve their goals without any commitments to behave in a responsible manner (Williamson 1994). In social governance, coordinating and safeguarding are facilitated by social mechanisms rather than authority, bureaucratic rules and other formal governance mechanisms (Jones *et al.* 1997). Thus, the current study proposed that in distributed agile development projects, safeguarding can be achieved through governance mechanisms such as reputation, collective sanctions and restricted access.

Coordination challenges are overcome through macroculture and restricted access (Jones *et al.* 1997). For example, restricted access means that for distributed agile development projects, a business representative is nominated as a point of contact for the duration of the project. Thus, the IT people only interact with this person and not any other potential user. The reduction of available partners results in lower coordination costs and more frequent interaction between the distributed project teams and the business representative (Jones *et al.* 1997, Feller *et al.* 2008, Sagers 2004). Macroculture also reduces coordination costs not only through the establishment of a common language among the IT people but also through the specification of shared tacit rules regarding how to behave with the business representative and what to expect from this person during the development process (Jones *et al.* 1997, Feller *et al.* 2008). Similarly, traditional software projects with their top-down, hierarchical approach to development and formal specifications of requirements (i.e. contracts) have a binding expectation to the extent that such expectation serves as a safeguard against opportunism in the relationship (Yadav *et al.* 2007).

Table 2. 6 Distributed Agile Development Project Governance Mechanisms

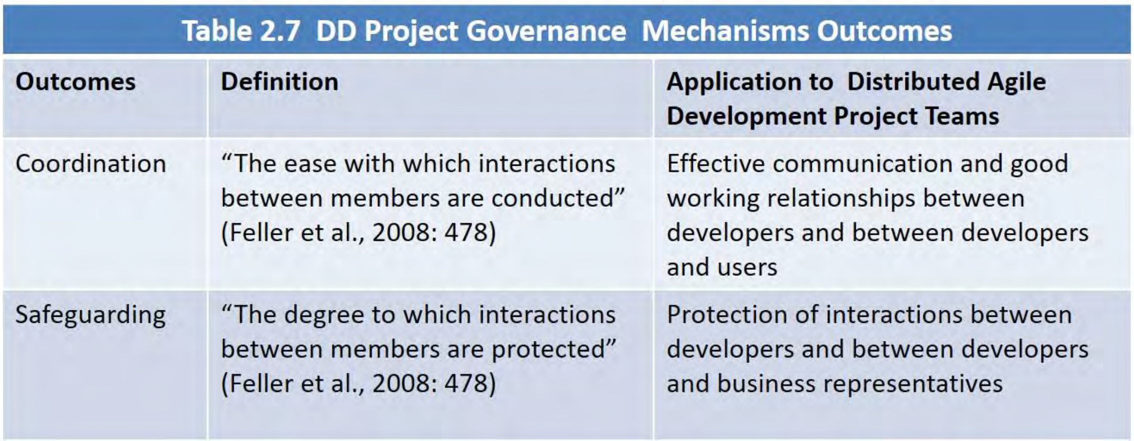


Table 2.7 DD Project Governance Mechanisms Outcomes

Outcomes	Definition	Application to Distributed Agile Development Project Teams
Coordination	"The ease with which interactions between members are conducted" (Feller et al., 2008: 478)	Effective communication and good working relationships between developers and between developers and users
Safeguarding	"The degree to which interactions between members are protected" (Feller et al., 2008: 478)	Protection of interactions between developers and between developers and business representatives

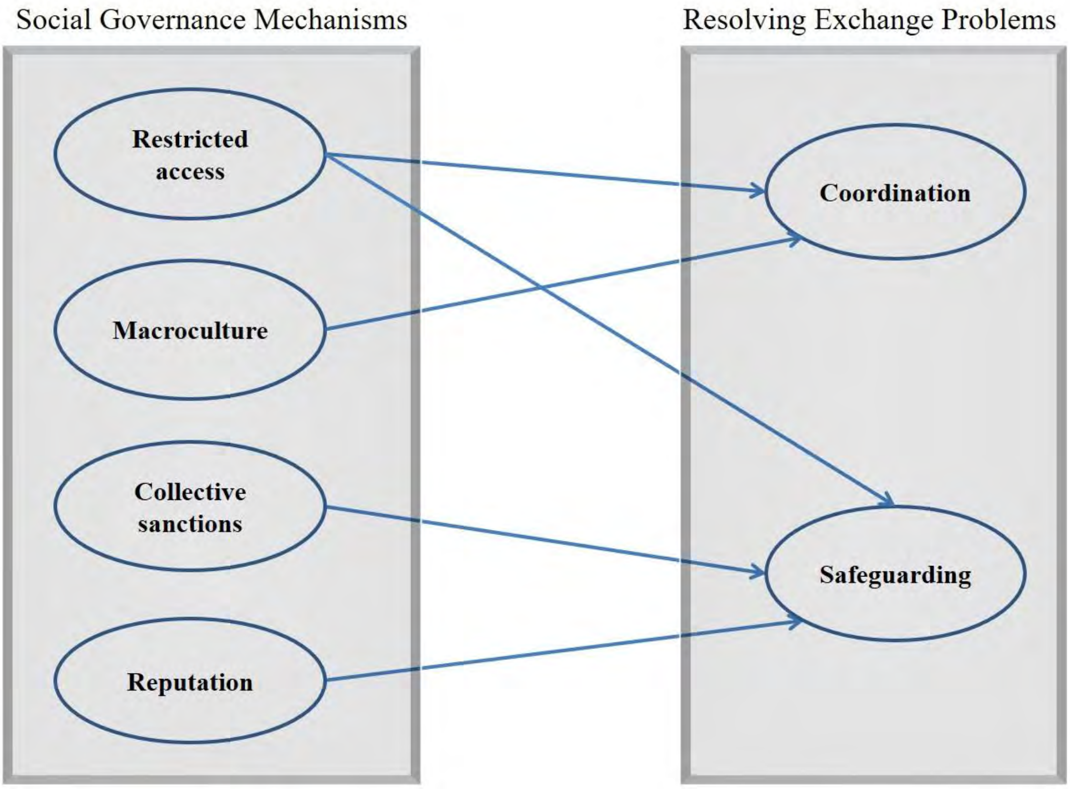
In summary, the study proposed that exchange challenges can be addressed by social governance mechanisms. In addition, social governance mechanisms are able to effectively govern distributed agile development projects through coordinating and safeguarding the project team exchanges. The application of coordination and safeguarding in distributed agile development project context is presented in Table 2.6 above.

2.6 Research Model

The current study proposed that distributed agile development projects can be conceptualised using social governance mechanisms because the essential characteristics of social structures (where developers work in a complex and uncertain environment) are present in distributed agile development projects. In addition, the exchange conditions of high coordination, high adoption and high safeguarding exist in distributed agile development projects.

In light of the above discussion, the formal model of NG proposed by Jones *et al.* (1997) was adapted for this study. To summarise, the adapted model proposes that applying social governance mechanisms to distributed agile development projects will lead to better coordination of efforts and better safeguarding of exchanges among the project teams.

Figure 2. 3 Adoptive Governance in Distributed Agile Development Projects



Adapted from Jones *et al.* (1997)

The current research study investigated how NG’s four social mechanisms (macroculture, restricted access, collective sanctions and reputation) could facilitate coordination and safeguarding of exchanges across distributed agile development teams. Indeed, the research intention was to understand and investigate contemporary phenomena in a real-life context. Based on this, I adopted a case study research design guided by a theoretical research model adapted from Jones *et al.* (1997) (Figure 2.3), which is discussed in more detail in the next chapter.

Chapter 3 Research Method

3.1 Introduction

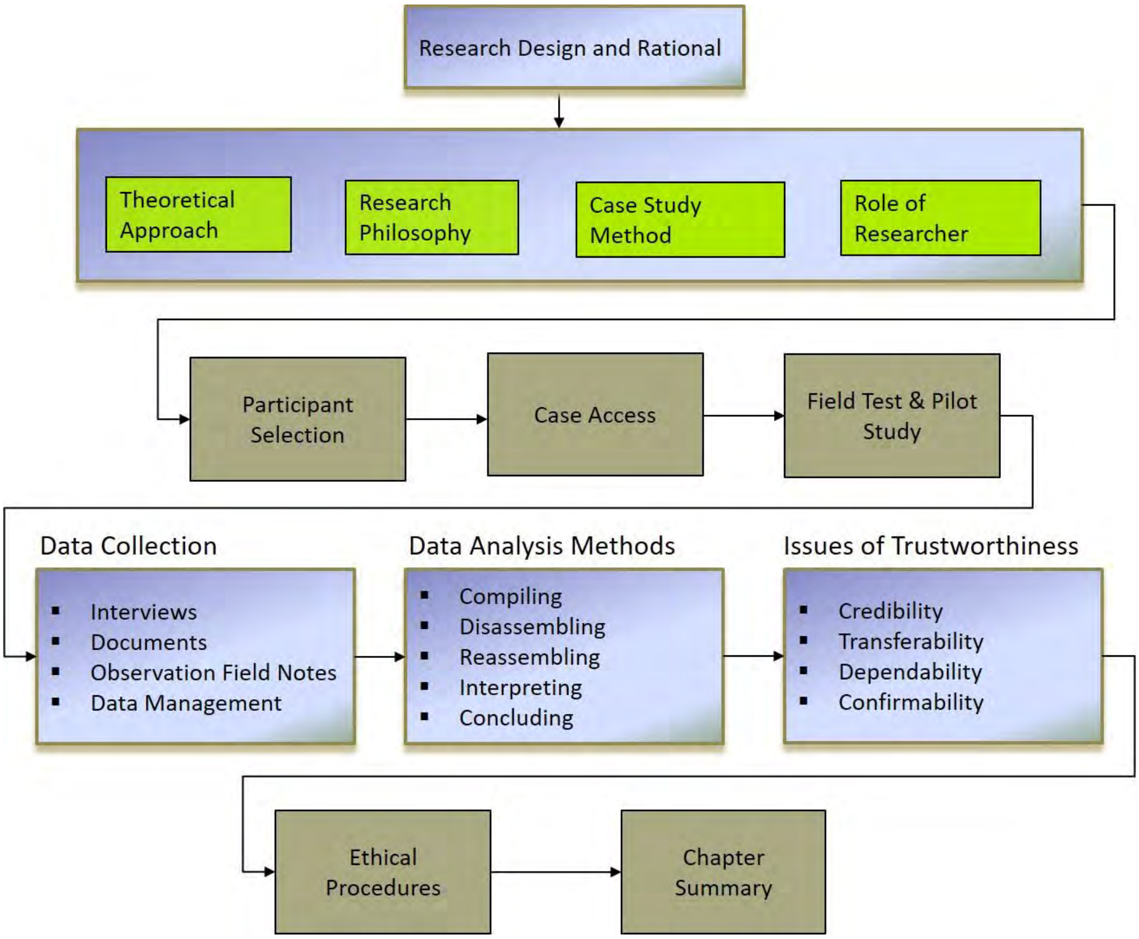
This study aimed to explore and develop a governance approach for distributed agile development projects in an original manner, through analysing. It also aimed to find other mechanisms and practices that are used by distributed agile development project teams in the same context. The current and previous studies seemed inadequate when explaining this phenomenon with very little empirical studies; therefore theory building from case study research was particularly appropriate for this study (Silverman 2000). The proposed model of social governance by Jones *et al.* (1997) was adopted for this study. The study also aimed to find additional relationships in the Jones *et al.* (1997) model and the effective practices to improve coordination and safeguarding in DD projects. Chapter 2 provided an extensive and critical review of the literature that led to knowing the importance of social governance mechanisms for governance of distributed agile development projects. NG theory was then used as a theoretical lens to determine and understand distributed agile development project governance mechanisms.

The current study aimed to investigate the research problem of *"How distributed development projects can be governed effectively*?" Yin (2009) indicated that defining the research questions is probably the most important step in a research study. Three research questions were developed for the purpose of this study (See section 1.3).

This chapter outlines the various components of the adopted methodology and articulates the philosophical foundations within which the current study rests. The chapter begins with the research design and underlying philosophy (Section 3.2). This section outlines the theoretical method, the chosen research methodology and case study design justification as well as the research questions which will be the focus of this study. Subsequently, Section 3.3 presents the participant selection procedure and sampling methods. The case access is then discussed in Section 3.4 followed by field notes and pilot case study (Section 3.5). Section 3.6 and 3.7 respectively presents data collection and data management and analysis methods for this research followed by issues of trustworthiness (Section 3.8) and ethical procedures (Section 3.9).

Finally, the chapter summary is presented in Section 3.10. Figure 3.1 summarises the structure of this chapter.

Figure 3. 1 The Structure of Chapter 3



3.2 The Research Design and Rationale

Selection of research methodology and methods for this study were guided by the following factors suggested by Creswell (1998):

- the nature of the research questions
- the exploratory nature of the study
- the need for a detailed view of the situation or phenomenon being studied
- the approach that would accommodate the study of individuals in their natural setting

With these factors in mind, I chose a qualitative case study involving methodological triangulation (Merriam 2009), with the gathering of data in the form of participant interviews, observations, and documentary evidence. This design strategy was selected because it offered a greater capacity for understanding how DD projects can be governed effectively.

The design for this case study was strengthened using multiple methods (e.g., Yin 2009). The research design was based on using certain fundamental elements linked to the research questions (e.g. Cassell & Symon 2004). These elements included the theoretical tradition of inquiry that undergirds this research, the research population, the methods of data collection and analysis, the narrative report structure, quality and ethics related issues, understanding the researcher's role as well as addressing the biases of the researcher and the protection of all participants.

3.2.1 Theoretical Approach

To conduct empirical research, there is a choice of either qualitative or quantitative method. Qualitative research takes a more holistic approach to the research and studies a phenomenon in its context (Creswell 2008). Qualitative research offers an exploratory approach typically used when the researcher does not fully understand the variables to be examined and when methodological traditions are requisite for providing an intensive description and analysis of a phenomenon or social unit (Creswell 2007). Qualitative research allows for greater attention to nuance, interdependencies, idiosyncrasies and context, which helps create a deeper understanding and description of complex systems (Patton 1990).

It is argued that qualitative methods go beyond the measurement of observable behaviour (i.e. the "what") and try to understand the meanings and beliefs underlying the observable behaviour (i.e. the "why" and "how") (Marschan-Piekkari & Welch 2004). In the current study, I tried to understand and interpret how governance mechanisms are being used in distributed agile development projects. I tended to observe the phenomena of governance in distributed agile development projects, asked questions and interacted with professional people in depth at the field site. In addition, the research problem and research questions were posed as "how" questions and I had no control over the behavioural events of the respondent and the study focussed on contemporary events (Yin 2003). The focus of the study was the governance of distributed agile development projects by applying governance mechanisms in order to achieve enhanced coordination and safeguarding exchanges within distributed agile development project teams. These were not one off events (Yin 2003). Therefore, I conducted my research using a qualitative research method.

Being a qualitative case study, I conducted an in-depth description and analysis (Creswell 2007, Yin 2009) of social governance mechanisms in distributed agile development projects to enhance coordination and safeguarding of exchanges among the project teams. Additionally, I did not seek to establish a causal relationship between variables, if any, which is typical of an explanatory method (Marschan-Piekkari & Welch 2004).

An exploratory case study is particularly appropriate when the field of study is still evolving (Walsham 1993). The essence of exploratory research is to investigate and provide a detailed study in order to shed different light on a problem to uncover new knowledge (Creswell 2007, Goulding 2002, Yin 2009), which is consistent with the current research. There have not been any significant reports suggesting governance mechanisms as an answer to the challenges of distributed agile development projects. The impact and application of social governance mechanisms by the project teams must be studied with caution because they are highly sensitive and controversial. Researching this issue with caution entailed respecting the views of the participants and using multiple sources to gather rich information. Furthermore, I selected the exploratory research approach because it allowed me to have a better understanding of the research problem regarding how DD projects can be governed effectively. In using this exploratory method, I was able to clarify the complex aspects of applying social governance mechanisms (restricted access, macroculture, collective sanctions and reputation) in distributed agile development projects and the impact of each of these mechanisms has on coordinating and safeguarding the project team’s interactions and exchanges (Walsham 1993).

3.2.2 Research Philosophy

The philosophical base of interpretive research is hermeneutics and phenomenology (Mumford *et al.* 1985). Philosophical Hermeneutics has a non-objective view of meaning and aims to understand what is involved in the process of understanding itself. It is argued that understanding is not an isolated activity but a very condition of being human; part of our experience of life. Interpretive studies generally attempt to understand phenomena through the meanings that people assign to them (Gallagher 1992). Interpretive methods of research in Information System (IS) are aimed at creating an understanding of the context of IS, and the process whereby IS influences and is influenced by the context (Walsham 1993). The current research was interested not only to study and describe the observable behaviour of the respondents but also aimed to understand their feelings, values, norms, interests, motivations and activities. Such specifications made the study more compatible with interpretivist ontology.

In the current study, I was directly involved in the process of data collection and analysis, which makes it more compatible with the qualitative and interpretivist case study approach (Walsham 1995a, Creswell 1998, Klein & Myers 1999, Morgan & Smircich 1980, Morse 1994). The interpretivist ontology provided an opportunity for me to get a deep insight into the problem under study because "[a]n interpretive explanation documents the [participant's] point of view and translates it into a form that is intelligible to readers" (Neuman 1997, p. 72). Indeed, this interpretive research made it possible to present my own constructions as well as those of all the participants (Guba & Lincoln 1994, Walsham 1995a).

3.2.3 Case Study Method

The case study methodology was determined to be the most applicable for investigating social governance mechanism's impact on coordinating and safeguarding exchanges in distributed agile development context and the relationship between social mechanisms. Yin (2009) suggested that case studies are used for research on a contemporary phenomenon within some real-life context and for exploratory purposes when "how" or "why" questions are posed. Merriam (2009) added that a case study approach is often selected as a means to reveal knowledge about a phenomenon that may be missed by standard statistical methods, especially when such a phenomenon is counterintuitive, nonobvious or infrequent.

According to Crotty (1998), all research is driven by distinct methodological assumptions that originate in the theoretical underpinnings of a certain research approach. The epistemology and theoretical perspective outlined in this study were grounded in the constructionist framework, and the methodology used allowed for a more accurate description of the nature of social governance mechanism's relationships in a distributed agile development environment, as well as a detailed account of the project teams' perceptions regarding the impact of social governance mechanism's impact on coordination and safeguarding exchanges (Lincoln & Guba 1985). Accordingly, this methodology offered an emergent design that allowed for understanding to evolve through the research process while providing the flexibility to pursue new avenues of inquiry when needed (Patton 2002).

I adopted a qualitative research design for the current study by employing a single case study that was consistent with the interpretivist ontology of the research and made it possible for this study to create new lines of thinking by the emergence of fresh perspectives and contradictions. The type of case study chosen was based on the richness of the competitive propositions in theories related to the topic of the study (Yin 2009).

The case study method allowed for the systematic gathering of evidence from respondents and inscription devices in a cross-sectional manner to answer the research questions (Neuman 2003). This approach provided an opportunity for a detailed contextual analysis of distributed agile development projects' social interaction and underlying governance mechanisms with the aim of building theory for distributed agile development governance.

In addition, the case study approach is an appropriate method for dealing with a complex process involving institutional, organisational and strategic issues such as effective governance of distributed agile development projects. The case study approach provided the opportunity for me to conduct semi-structured interviews and use questions in interviews that were able to be tailored to open avenues of exploration that seemed to yield information relevant for the topic being studied (Bailey 1987). Indeed, among various methods used in collecting empirical data in case studies the most important way was in-depth, open-ended interviews.

The case study protocol is referred to as a major component in asserting the trustworthiness of the case study research (Yin 1994). A case study protocol contains procedures and general rules that should be followed in using the instrument and needs to be created prior to the data collection (Yin 2003). The development of the case study protocol gave me the opportunity to detail in advance the procedures and requirements to be followed during data collection (Perry 1998). It also provided direction for me to improve the reliability of my research findings.

Following Yin's (1994) recommendation, I developed a case study protocol (Appendix A). While I prepared the topics and protocol for interviews in advance, the actual questions were not specified in advance but depended on the context of interviews (Yin 2003). This was very useful, because the interviewees provided crucial insights. Thus, the case study approach represented a powerful and particularly timely method of scientific inquiry and it permitted a deeper understanding of soft variables and key relationships, which was compatible with this study.

3.2.4 Roles of Researcher

Qualitative methods involve the process of collecting, analysing, interpreting and writing the results of a study (Creswell 2007). As the primary instrument for data collection (Creswell 2007, Yin 2009 and 2011), I performed the entire research throughout all stages (i.e., data collection, data analysis and report writing) by undertaking the following activities:

- adapting and refining the research design
- identifying the site of the study, interview protocol and data collection tools
- visiting the project site to collect any relevant data, which includes performing interviews with interviewees
- seeking the necessary consent and permission from UNSW and the organisation under study (Ocean Group) to collect documents and conduct interviews as well as ask for assistance with identifying a pool of possible participants
- using telephone calls, emails, and letters to contact participants
- being open to making personal contact where necessary
- conducting all the audio tape recordings and transcriptions of interviews, analysis, and reporting in this research, and
- making notes during the interviews and utilising them during the data analysis and interpretation

As is characteristic of qualitative research, I was the primary instrument for data collection in this study (Creswell 2007). Therefore, the potential for researcher bias existed (Creswell 2007, Yin 2011). It is noted that "Investigators need to explain their biases, dispositions, and assumptions regarding the research to be undertaken" (Merriam 2009, p. 219). Furthermore, Merriam (2009) argued that such a clarification allows the reader to better understand how the individual researcher might have arrived at a particular interpretation of the data. Therefore, specific strategies that I used to enhance the credibility and trustworthiness of this study are explained in more detail in section 3.8 later in this chapter.

This role required that the researcher be skilful in interviewing participants, very observant about the research environment and able to interpret body language and other nonverbal cues, in order to capture significant or pertinent data for this research (Creswell 2007, Goulding 2002, Polkinghorne 2005). I had extensive experience and skills in observing and interviewing individuals based on my previous work experience as a project manager. In order to gather data exhaustively, I acquired and triangulated different data collection instruments such as documentary sources, interviews and observational field notes (Creswell 2007, Goulding 2002, Polkinghorne 2005). These instruments are explained in section 3.6. In order to ensure adequacy of data, I used a broad range of sampling and data collection strategies (Morrow 2005). I alone collected and analysed the data at all times.

For the data collection part of this study, I assumed the role of participant observer. Merriam (2009) defined this role as one in which my observation activities are known to the group, but my participation in the group is secondary to the role of information gatherer. Being an active participant in the learning process of the study involved listening, not as an expert but as a curious student of learning. The research process was based on the tenets of trust, mutual respect, integrity and partnership in learning. I observed what was related to the physical setting, availability of communication tools, different ways of interactions and communications between the project teams across locations and the environment within which project activities took place. Observation generated insight and better understanding of the phenomenon under study.

Researchers can become a source of bias when they are the main instrument for data collection (Yin 2011). I, as the researcher of this study, was an outsider to the population being studied because I had no direct contact or relationship with the organisation which was to be studied. Usually, the possibility of bias would come from the researcher’s proximity to the environment or population (Goulding 2002); however, this did not apply to this study. The researcher bias can be attributed to the researcher's background, motive for undertaking the research and personal filters (Yin 2011).

Consequently, in this study, I employed certain measures to address the possibility of subjectivity. I maintained a very high level of consciousness about these biases and tried to limit these biases where possible and remain objective throughout the research. I also used peer reviews and team checking as steps to enhance the credibility of the study results (Creswell 2007, Goulding 2002). Feedback comments from selected participants were integrated into the report. The steps mentioned earlier were used to assure trustworthiness such as documenting the process of category development to foster openness, thereby minimising any possible bias or subjectivity. The data from the team meetings and the minutes were constructed independently of this study which enabled me to analyse them independently (Yin 2010). The connections and analysis made were solely from my viewpoint.

3.3 Participant Selection

Selection in qualitative research is the selection of participants and pertinent documents for a given research study (Polkinghorne 2005). The participants and documents related to a given study must (be able to) inform the researcher in order to increase understanding of the concept being studied (Polkinghorne 2005). The use of participants and documents pertinent to the research such as IT structure, the project team social contract and the project team set of values enabled me to triangulate the findings. Furthermore, it is noted that the sample selected should "establish a school for the domain that the case concerns" (Flyvbjerg 2006, p. 230). Information collected from the sample came from transcripts of semi-structured interviews, project artefacts, emails and other project-related documentation.

Qualitative selection procedure can provide the opportunity to select and examine observations of generic processes which are key to our understanding of new or existing theory about the phenomenon being studied (Miles & Huberman 1994). The implications are that theory will drive the selection of these cases, and also that the careful examination of the cases may lead to elaboration or reformulation of theory. Consequently, purposefully selecting participants is a vital decision point in a qualitative study (Creswell 2007, Patton 1990, Eisenhard 1989).

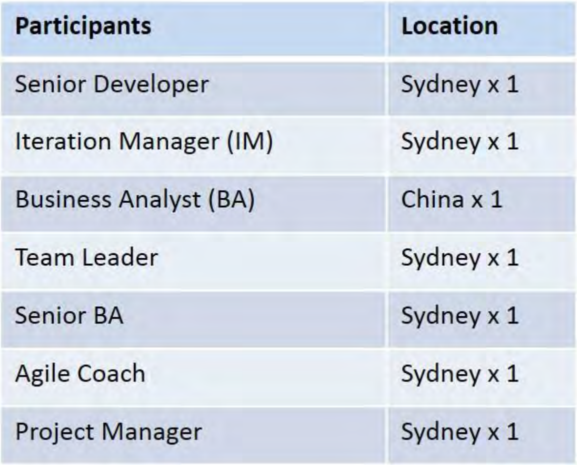
Different types of selection techniques was noted in the literature (Merriam 2009), where among these techniques, the two selected for this research project were purposeful and snowball. “Purposeful selection is based on the assumption that the investigator wants to discover, understand, and gain insight and therefore must select a sample from which the most can be learned” (Merriam 2009, p. 77).

The selection strategy employed in this study was purposive, which is a preferred selection method for the qualitative research (Creswell 2009, Ritchie *et al.* 2003, Perry 1998, Patton 1990). Purposeful selection was applied by identifying specific roles in the project, such as the project manager, who could speak about the application of social governance mechanisms and the challenges in the project from a different point of view. This selection approach was used as a starting point. The second selection approach, snowball, followed semi-structured interviews with the first sampling set. According to Miles and Huberman (1994, p. 28), snowball selection “identifies cases of interest from people who know what cases are information rich,” thus inviting participants for whom the research question will be meaningful (Smith *et al.* 2013). The snowball selection allowed me to access persons in other roles on the project team, not previously identified by me, who had useful insight to the success and challenges of the project, such as the agile coach.

Upon receiving UNSW Human Research Ethics Advisory’s (HREA) approval for data collection, I notified the project iteration manager (IM) as the main contact person in an email requesting a meeting to formulate an agreed strategy for conducting the research on the project site and introduction to the project stakeholders. Attached to this email was a letter of invitation to participate in research with the approved HREA (Appendix C).

The selection of the interview participants began with a letter explaining the research project, which was sent to the iteration manager within the selected project. The letter also included an invitation to select individuals on the team who are available and could be contacted for an interview. I requested permission to interview at least the following five roles: the project manager, the business analyst, the iteration manager, the team leader and the developer. All of the selected participants interviewed were based in Sydney. However, I schedule the interviews so I can also arrange an interview with the business analyst that was based in China but visiting Sydney at the time of my data collection. I was the able to schedule a face to face interview with the business analyst from China team in Sydney. In addition to "their special experience and competence" (Merriam 2009, p. 77), the purpose behind selecting these specific individuals was to gather a broad perspective on the experiences across different roles to better understand how they used social governance mechanisms and their impact on coordinating and protecting the project teams' exchanges across locations.

Table 3. 1 Participants



Participants	Location
Senior Developer	Sydney x 1
Iteration Manager (IM)	Sydney x 1
Business Analyst (BA)	China x 1
Team Leader	Sydney x 1
Senior BA	Sydney x 1
Agile Coach	Sydney x 1
Project Manager	Sydney x 1

Once the participants were identified (Table 3.1), I contacted each participant by email. The email introduced me as the researcher, provided the purpose of the study and a brief explanation of the research context, time required for interview and an explanation about how the participant was selected as a potential participant in the interview process. It also stated that the participation was entirely voluntary and no compensation would be given. Seven participants agreed to be interviewed, at which point the consent form was sent out to each participant for their review. Each participant could then either accept or reject participation in the project. Upon receipt of each potential participant's agreement to sign the consent form and to participate, the researcher began scheduling interview times that wouldn't impact the participants normal work schedule. The selection of participants and the specifics about how the project case was selected (selection criteria) are discussed in Chapter 4.

It is important to note that the preferred selection criterion in this tradition is experience and not representativeness. The qualitative analysis is not about statistical significance and therefore does not require a sufficient scale for samples (Denzin & Lincoln 2003). Given the purpose of evaluation in a qualitative study, the selection size for this study was large enough for the purpose of credibility but small enough to permit detail and depth for each unit of analysis (Patton 1987).

The unit of analysis is a critical factor in the case study (Myers 1997). It is typically a system of action rather than an individual or group of individuals (Walsham 1995a, Myers 1997). The case studies tend to be selective, focussing on one or two issues that are fundamental to understanding the system being examined. Based on the nature of the research question, the unit of analysis for this study was the project team and because the data was collected from individuals, therefore the unit of data collection was the project team members.

3.4 Case Access

This study took place in the case study company, which is referred to with the pseudonym 'Ocean Group'. The Ocean group building was located in the CBD, which was easily accessible to me. Prior to the start of the study, the iteration manager of the project granted written permission to conduct the research at the site. I provided the superintendent with written information about the goals and intention of the study, the selection of participants, the methods used and the manner in which ethical considerations, including consent, would be addressed.

Therefore, I paid particular attention to this element of the research because I was required to have repeated access to the case study organisation and the organisation was concerned about confidentiality and safeguarding its reputation in permitting the access. I identified the iteration manager for the related project case as the main gatekeeper to the organisational research that could introduce and lead me to useful informants. Then I quickly established other significant people in the organisation such as the team leader for the related project case.

I followed a set of six criteria proposed by Miles and Huberman (1994) to evaluate the sampling and selection of the case for this study. I found the suggested six criteria helpful when evaluating a sampling plan. They address the relevancy of the sampling in connection with the conceptual framework, the chance to study the phenomena of interest, the possibility to describe and explain, feasibility of the plan in terms of time and money and finally, ethics issues. These criteria are applied to this study as follows:

1. The sampling strategy was relevant to the research conceptual framework and the research questions addressed by the study.

2. The selected case was able to generate rich information on social governance mechanism's application and their impacts on safeguarding and coordinating of project teams' activities across borders, which is conceptually important for this study.

3. The selected case enhanced the generalisability of the findings. In this qualitative study, I was concerned with analytic generalisability rather than statistical power to make statements about a general population on the basis of the selected case.

4. The selected case produced believable descriptions/explanations and provided a convincing explanation of what was observed (in the sense of being true to real life).

5. I also considered the ethical nature of the relationship between the researcher and informants. All participants signed a consent form and agreed with the data collection method.

6. I considered the feasibility of the sampling plan in terms of the resource costs of money and time, the practical issues of accessibility and whether the sampling strategy was compatible with my work style. Other important points to take into consideration here were my competencies as the researcher for feasibility, for example, in terms of linguistic and communication skills, ability to relate to participants and their experiences and my or the participant's capacity to cope with the circumstances under which data collection took place.

3.5 Field Test & Pilot Study

The following sections discusses the pilot case studies rational and provides details about the pilot case studies under taken.

3.5.1 Rational for conducting the pilot

I used the pilot cases to conduct pilot tests using data gathering methods. The pilot case studies provided me an opportunity to refine the data collection plans and test out the case study protocol (Perry 1998, Yin 2013). The pilot studies also allowed the problematic areas to be uncovered and corrected which made the result more trustworthy and reliable.

It is argued that the selection of cases should be based on criteria that is consistent with the research problem (Ghauri 2004). Patton (1990) outlines that the aim of selecting the cases purposively is that the cases should be information rich. Information rich cases are "those from which one can learn a great deal about issues of central importance to the purpose of the research" (Patton 1990, p.169). In addition, case study research relies on theoretical sampling as opposed to statistical sampling (Eisenhardt 1989). Hence, I selected the cases in order to replicate previous cases or to extend emergent theory. The random selection of cases was considered unnecessary and Eisenhardt (1989) advises that this is not a preferable strategy in a case like this.

The general principles for the selection require the case to be representative, comparable and manageable (Miles & Huberman 1994). I considered several factors in the selection of the case for governance of distributed agile development projects. The first factor was the accessibility to the organisation, respondents, documents and information related to the case. This was done by using personal networks and contacts, by choosing an organisation that the research had potential relevance and benefit to them, by ensuring that time and resources requested were minimal and that the researcher had a good reputation with the company through previous research and work experience. In this way, the organisation was willing to support the study. The second factor was to have a well distributed project across locations. This could be across states, or across countries or both. In the current study, the project team were distributed across states and across countries. Another factor that I considered for case selection was derived from the theory. Importantly for the NG theory model to be applicable, the case was required to meet the four preconditions for the NG theory (Table 2.5 and Section 4.3) including demand uncertainty, execution of complex tasks under time pressure, customised asset-specific exchanges and frequent exchanges (Jones *et al.* 1997).

Based on the above two factors, I selected the pilot cases for theoretical, not statistical, reasons, for extending emergent theory and to enable me to answer the research questions in a more efficient way (Patton 1990). Indeed, the goal of this theoretical selection was to choose cases which were more likely to replicate or extend the emergent theory.

I applied the data gathering tools to the selected pilot cases to determine whether the planned timeline was feasible and whether or not the interview questions and topics to be discussed were appropriate and effective, and then based on the results I made the required adjustments (Yin 1994). In the pilot cases, the context of the cases differed in their experience of using development methodology (agile/non-agile). For example, some of them were junior users and some were experienced distributed agile development project teams using agile methodology. Also the cases were different based on the way the project teams were distributed across geography (across states/ across countries).

I conducted pilot studies to provide insight into the key concepts and perspectives of the people being studied, as well as a means to determine the effectiveness of the instruments that were employed in the larger research study and to test and refine the interview questions (Yin 2011). I used the interview sessions in the pilot studies to model the interviews for the main study, and identified ways to maintain a flow for the sequence of questions that can comfortably be used in the main study (Yin 2009).

3.5.2 Pilot studies undertaken

Three pilot studies were held with company A, company B and company C. It involved three interviews from three different project cases to test the pilot interview protocol. An early version of the protocol is listed in Appendix N. All of the participants answered their questions and made useful comments about the flow and consistency of the questions. In general, the participants were satisfied with the interview protocol, which I had developed through an iterative and consultative process. However, the participants were not clear about some of the construct definitions such as safeguarding, or differences between values and norms. I changed and added some of the pilot questions and tried to use clear examples to clarify the constructs in more details for the participants in the interview protocol. In this study the following pilots were undertaken:

Pilot 1: Company A

Company A was a consulting organization with project teams distributed over Sydney (Australia), Hong Kong (China) and Dalian (China). I interviewed the project coordinator in Sydney. The project teams in Sydney and Hong Kong were senior agile team with years of experience of working with agile methodology where the Dalian team was junior agile team. The objective for the pilot 1 was to first test the protocol, run the questions, get some confidence, practice my interviewing skills and understand the timing. Also I tried to understand the capacity of the interviewee to answer the questions and how he would answer the questions. Revisions made to protocol for pilot to included. Questions were revised, timing was revised and I became more familiar with the data collection process (Table 3.2).

Pilot 2: Company B

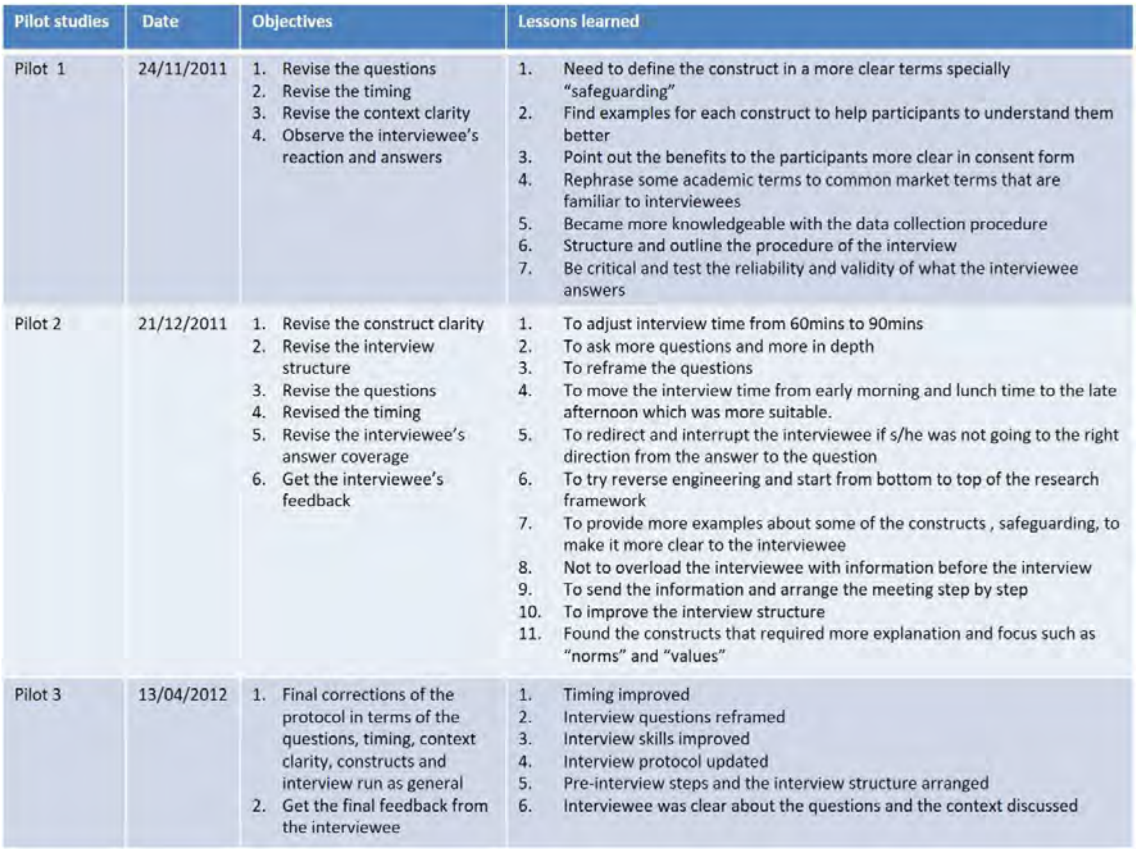
Company B was an insurance organisation with project teams distributed over Sydney (NSW) and Brisbane (Qld). I interviewed the project manager. The project teams in Sydney and Brisbane were senior agile teams with years of experience using agile methodology and practices. The objective for pilot 2 was to revise the construct clarity, interview structure, and interviewee's answers and to get more feedback from the interviewee. Revisions made to modify the interview time, reframe the questions, and clarify some of the constructs and to run the interview smoothly (Table 3.2).

Pilot 3: Company C

Company C was a financial organization with project teams distributed over Sydney (NSW), Melbourne (Vic) and Gurgaon (India). I interviewed the senior developer from Sydney team. The project teams in Sydney and Gurgaon were senior agile developers and the Melbourne team had average experience working with agile methodology. The objective for pilot 3 was the final run of the protocol for final corrections in protocol. The protocol ran successfully and participants indicated that the questions were clear and the timing was fine. At the end of pilot 3 study, I made the final revision to the protocol (Table 3.2).

Since the context of the interview could affect the flow and consistency of the interview questions, I considered the comments made by the pilot study participants in preparation towards conducting the interviews. Table 3.2 provides more details about the pilot cases and the lessons learned through the pilot studies.

Table 3. 2 Pilot Cases

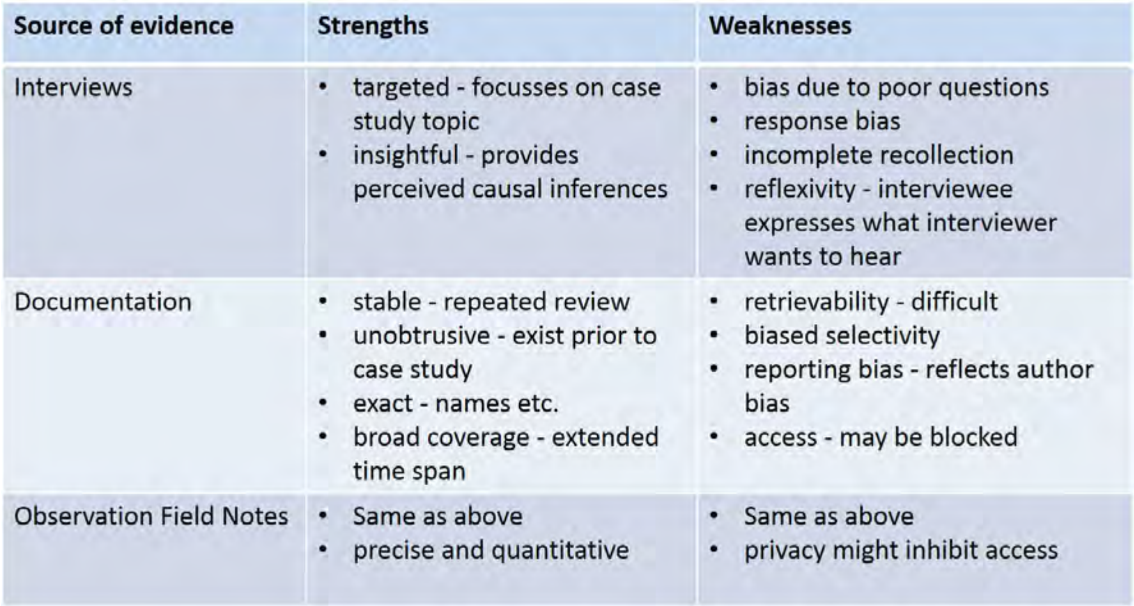


Pilot studies	Date	Objectives	Lessons learned
Pilot 1	24/11/2011	1. Revise the questions 2. Revise the timing 3. Revise the context clarity 4. Observe the interviewee's reaction and answers	1. Need to define the construct in a more clear terms specially "safeguarding" 2. Find examples for each construct to help participants to understand them better 3. Point out the benefits to the participants more clear in consent form 4. Rephrase some academic terms to common market terms that are familiar to interviewees 5. Became more knowledgeable with the data collection procedure 6. Structure and outline the procedure of the interview 7. Be critical and test the reliability and validity of what the interviewee answers
Pilot 2	21/12/2011	1. Revise the construct clarity 2. Revise the interview structure 3. Revise the questions 4. Revised the timing 5. Revise the interviewee's answer coverage 6. Get the interviewee's feedback	1. To adjust interview time from 60mins to 90mins 2. To ask more questions and more in depth 3. To reframe the questions 4. To move the interview time from early morning and lunch time to the late afternoon which was more suitable. 5. To redirect and interrupt the interviewee if s/he was not going to the right direction from the answer to the question 6. To try reverse engineering and start from bottom to top of the research framework 7. To provide more examples about some of the constructs , safeguarding, to make it more clear to the interviewee 8. Not to overload the interviewee with information before the interview 9. To send the information and arrange the meeting step by step 10. To improve the interview structure 11. Found the constructs that required more explanation and focus such as "norms" and "values"
Pilot 3	13/04/2012	1. Final corrections of the protocol in terms of the questions, timing, context clarity, constructs and interview run as general 2. Get the final feedback from the interviewee	1. Timing improved 2. Interview questions reframed 3. Interview skills improved 4. Interview protocol updated 5. Pre-interview steps and the interview structure arranged 6. Interviewee was clear about the questions and the context discussed

3.6 Data Collection

In this study, I considered multiple sources of data such as interview, document examination, gathering and study of organisational documents and observation field notes (Table 3.3). Some internal documents, including strategies, plans and evaluations, and web-based data from emails and intranet message posting were very valuable.

Table 3. 3 Types of Evidence



Source of evidence	Strengths	Weaknesses
Interviews	• targeted - focusses on case study topic • insightful - provides perceived causal inferences	• bias due to poor questions • response bias • incomplete recollection • reflexivity - interviewee expresses what interviewer wants to hear
Documentation	• stable - repeated review • unobtrusive - exist prior to case study • exact - names etc. • broad coverage - extended time span	• retrievability - difficult • biased selectivity • reporting bias - reflects author bias • access - may be blocked
Observation Field Notes	• Same as above • precise and quantitative	• Same as above • privacy might inhibit access

Yin (1994) and Kitzinger (1998)

3.6.1 Interviews

Interview was the primary mode for data collection in this research because a) case study research requires extensive interviews (Creswell 2007) and b) one of the most notable and primary sources of data collection in qualitative research is interview (Creswell 2007, McReynolds *et al.* 2001). Interviewing is “a process in which a researcher and participant engage in a conversation focused on questions related to a research study” (Merriam S. B. & Tisdell E. J. 2015, p.55) with the purpose of entering into the participant’s perspective (Patton 2002) to extract “a special kind of information” (Merriam 2009, p. 88). Hence, if the goal of interviewing is to capture the perspectives of participants, then it is to some degree, imperative (Hatch 2002, p. 97).

Literature, as reviewed in Chapter 2, provided the basis for the research questions and the subsequently developed interview questions. Content validity was established through multiple written records in the form of data team meeting minutes triangulated with a couple of interviews for each of the seven participants to discuss their perspective of application of social governance mechanisms in distributed agile development projects and their impact on coordinating and protecting their interactions across locations.

I preferred to use individual face-to-face interviews as the best way to obtain the data needed to answer the research questions, supplemented with the written record. This form of interview offered the respondents the psychological safety needed to freely air opinions and offer the best and trusted information. The use of the one on one interview method also gave me, as the researcher, the benefit of observing each respondent's body language in order to contextualise the information collected and clarify any possible ambiguities (Hall & Rist 1999). Although individual interviews have the disadvantage of being time consuming and costly, and carry the possibility of having very little useful information from hesitant participants (Creswell 2007, Hall & Rist 1999), the benefits still outweigh these disadvantages. For instance, the richness of detailed information that can be derived from an individual interview where the interviewee is safe to freely respond to questions provides the benefit of accurate information.

The interviews were semi-structured with prepared questions focussed on the research objectives in order to guide the process (Goulding 2002, Singleton & Straits 2005). In order to develop and pilot test a set of questions related to social governance mechanisms, I used guidelines from Creswell (2007) and Yin (2009). I employed a panel of experts to review the open ended questions to ensure that they focussed on the substance of the research problem (Rubin & Rubin 2005, Yin 2009). The content for the interview questions was extracted from social governance mechanisms literature. I developed the interview protocol (Appendix D) to ensure consistency and included setting up the interviews, maintaining confidentiality and checking for clarity after the interviews were finished (Boyce & Neale 2006). The interview protocol included the rules that guided the administration and implementation of the interviews to ensure consistency between interviews, and thus increased the trustworthiness of the findings. It contained open-ended questions about broad themes. The themes were the following:

(1) Background information about the company and the project

(2) Details on the software development approach used

(3) Interpersonal aspects of the software development method. The interpersonal aspects of relevance for this research related to the application of social governance mechanisms and their impacts on coordination and safeguarding exchanges among the project teams.

I used a standardised open-ended interview format in the initial interviews which allowed participants to contribute detailed information from their unique perspectives even while all were asked the same questions (Turner 2010). I collected data at the project site in a meeting room with limited distractions and at the participants' convenience. Interviews were conducted during lunch breaks or after business hours. Each interview began with receiving the signed consent form and thanking the participant for their involvement in the project, followed by a re-introduction and explanation of the research.

I informed the participant that the interview would be recorded both with an audio recording device and hand written notes. I audio recorded the initial and follow-up interviews with a digital recorder in order to capture information accurately and to safely keep the information for later retrieval, so that no details being expressed in the participant's interview answers would be lost. Then I coded each recorded interview to protect the interviewee's identity and information was downloaded and saved in a password protected file on a computer. All the participants agreed to be recorded but I listened very intently and made very scrupulous notes. An extensive account of an unrecorded interview was prepared immediately after the event in order to minimise information loss (Stake 1995). At the beginning of the follow-up interviews, I summarised the discussion to verify accuracy and to refresh the interviewee's memory about the context discussed before. I then asked about the participant's education background and their work experience in Ocean Group and before that.

Since the interview was designed as semi-structured, after the initial introduction I then asked each participant the questions prepared in the case study protocol (Appendix A). Starting each interview this way provided consistency in how the interviews began and allowed the participant to think about the selected project. Since participants played different roles, their personal perspectives about the project were important to capture. Each interview lasted approximately one hour to 90 minutes. Participants were also given an opportunity to express whatever was on their minds about the topic that was not covered by the interview questions. Upon conclusion, participants were thanked for their time and asked if it was acceptable to contact them again if other questions arose, or to review my notes to ensure their responses were captured and represented the way they intended. Each participant was also asked if there was anyone else on the project team who they felt might provide valuable insight to the research.

Any name collected, which was not already part of the initial list of potential participants, was reviewed with the iteration manager before being contacted by myself.

I attended a couple of meetings including Showcase meetings, Retrospectives, Stand up meetings, Homemade Jam meetings and the Developers' Forum (Section 3.6.3). The focus of accessing these meetings was to observe the interactions and exchanges among the project teams across geography rather than the specific content of the meetings. The content was not included as part of the detailed data analysis. Instead the collected information during these observations was used to review the application of social governance in the Lake Project and their impact on the way the Lake Project team's exchanges and interactions are coordinated and safeguarded across locations.

3.6.2 Documents

Data in the form of documents are potentially relevant to all case studies and it is argued that this should be part the explicit data collection plans for the study (Yin 2003). Documents can include emails, memoranda, agendas, announcements, minutes of meetings, progress reports and formal studies previously undertaken.

Document sources serve different purposes in qualitative research. Singleton and Straits (2005) observed that available data helped to refresh the memories of participants about their experiences because of the time lapsed. Documents can also act as additional sources of information that may be unknown to the participants. There were different types of available data to be gathered for this study including project status report, project core team activity report, organisation IT structure, the project team social contract, organisation and the project team values document. While documents are claimed to be of relevance to the data collection process, they may not always be accurate and lacking in bias (Creswell 2007, Merriam 2009). The literature suggests that these documents should not be used as literal recordings; instead they should be used to substantiate and augment the evidence collected from other sources (Yin 2003). If the documentary evidence is contradictory then further inquiry into the issue should be taken. I examined different types of data from various sources for this study and they confirmed the interviewees' claims.

Given the importance of this source of data, the reliance on documents was an explicit part of the research design for this study. I collected documentary evidence on the case study project and created a case study database in order to increase the trustworthiness of the study. I collected the data from the following documents during the data collection process:

- the team meeting minutes
- the project status report
- the performance report
- the IT structure
- retrospective plan
- the program structure
- the organisation web site
- the project team relational contract
- the organisation values

I coded and catalogued all documents to ensure confidentiality of the participants then added the document data to the database (Merriam 2009). I carefully reviewed the validity of the documents so as to avoid incorrect data being included in the database. One of the most important use of documents is to corroborate evidence gathered from other sources (Creswell 2007). Archival records were also useful in this study since they include project records and charts. I was precise in determining the origin of the records and their accuracy. In addition to interviews, empirical materials were gathered. These materials used in conjunction with the other sources of information to better understand the background and context around the project case.

3.6.3 Observational Field Notes

In this research, I used observational field notes (Appendix B) where necessary, in contributing to part of the findings and analysis of this research. Observational field notes are reliable sources of data but have some challenges (Creswell 2007, Merriam 2009, McReynolds *et al.* 2001). For instance, the mechanics of observing such as recording quotes accurately or remembering to take notes can make observational field note taking quite demanding (Creswell 2007). Although the interviews were recorded, observational field notes also preserved specific quotes and critical information during the interview (Appendix B). The different data sets and procedures used in the collection of data will be discussed in more detail in the next major section.

During data collection, I observed how distributed agile development project teams worked and interacted with each other. This direct observation helped me to understand the phenomena better so the data analysis was easier to conduct. Observations were conducted according to specific protocols in order to enhance the credibility and trustworthiness of this study. For example, I began the observation process by contacting the participants across four geographical locations (Sydney, Melbourne, Chengdu (China) and Brisbane) through the iteration manager. The IM introduced me by email as the researcher and explained the purpose of the study, why the project was chosen as the proposed case, the voluntary nature of their participation and the confidentiality of their identity. Then they were informed by me and the iteration manager that I would randomly be present in some of the project meetings as an observer.

I sought permission from the iteration manager to observe the project teams' interactions and exchanges across geographical boundaries. The date, place and time of observations were then confirmed with the iteration manager. The observation took place through my participation in the following four different project team gatherings across locations. In general the application of social governance mechanisms and their impact on protecting and coordinating the project team' interactions were observed:

- ***Daily Stand ups***; every morning at 12pm Sydney time, the teams met online where all the core project teams participate in a status update. This status allowed the teams to know about potential challenges as well as coordinating efforts to resolve difficult and time consuming issues. It was a great way to share info with the team during iteration across locations.
- ***Business analyst forum***; where all business analysts met across locations and interacted through a communication tool to discuss different issues, to give insights to any problems raised during the meeting time and to share information with other BAs for further action

- ***Technical forum***; where all developers gathered across locations and started to interact through Skype/ teleconferencing to check at the end of the day what they had done and discuss different technical issues. Each forum session ran for 20 to 30 minutes.
- ***Retrospectives***; every fortnight at the end of each iteration, a Retrospective meeting was held where everyone in the team across locations had their say as to how they thought project was going, how the team was performing, to discuss the issues and what could had been done to improve in future iterations.
- ***Showcase meetings;*** where the project team members presented the work they had done so far for the iteration in front of the stakeholders. This facilitated early feedback, reduced last minute surprises and ensured the development team was building the right product based on the business requirements. It was indeed a status update for senior managers of the project.
- ***Homemade Jam meetings***; every three weeks similar to Showcase but this was exclusively for any technical team members of the project organisation-wide to meet up and to interact with other technical team members that had similar interests. Representatives from all different IT teams in Ocean Group in Sydney, Chengdu (China) and Melbourne were invited. The goal was to socialise the project IT teams in the organisation to get their input and ideas. The reason for this meet up was that once this product is finished, it was going to go to one of these teams for maintenance. This facilitated the technical project team members to learn about techniques, practices and strategies used in this particular project. In addition, having regular Homemade Jam meetings enhanced the coordination in the maintenance phase of the project.
- ***IPM (Iteration Planning Meeting);*** it was rolled into the same meeting as the Retrospective - the project team members had a Retrospective and afterwards they had the planning meeting to figure out what they wanted to do for the next iteration. It helped in a way that the team members had reflected on what they want to do better with a mindset that opens the ways to plan the next iteration more of what they wanted to achieve.

I observed the frequency of interaction among project teams, the communication tools used to coordinate these exchanges, the participants' behaviours, the physical environmental settings for where the meetings were conducted, the transparency of the project teams' activities and the trust level in the teams' interactions across locations, the content of the exchanges and subtle factors such as nonverbal communication. Field notes and researcher reflections were recorded on a notebook pad secured to a clipboard. Once the observation was concluded, I thanked the participants before leaving the study site.

The following table summarises the case company visits and the type of data collection used at the time:

Table 3. 4 The Case Company Visits

What was done	Date	Time	Duration	Who attended	Location of Attendees
Interview 1	08/06/2012	10:00 am	2 hrs	Senior Developer	Sydney
Interview 2 (interview suspended due to interviewee time constraint)	27/06/2012	3:00 pm	50 mins	Project Manager	Sydney
Interview 3 (interview suspended due to interviewee time constraint)	11/07/2012	10:00 am	45 mins	Iteration Manager	Sydney
Interview 4	25/07/2012	2:30 pm	1:50 mins	Senior Business Analyst	Sydney
Interview 5 (interview suspended due to interviewee time constraint)	15/08/2012	10:00 am	1 hr	Team Leader	Sydney
Interview 6	11/09/2012	12:00 pm	1:50 mins	Agile Coach	Sydney
Interview 7	17/09/2012	10:00 am	1:45 mins	Business Analyst	China – visiting Sydney
Continuation of interview 2	28/06/2012	3:00 pm	45 mins	Project Manager	Sydney
Continuation of interview 3	18/07/2012	3:00 pm	1 hr	Iteration Manager	Sydney
Continuation of interview 5	17/08/2012	1:30 pm	55 mins	Team Leader	Sydney
Follow up interview 6	27/09/2012	10:00 am	30 mins	Agile Coach	Sydney
Follow up interview 7	13/09/2012	2:00 pm	35 mins	Business Analyst	China – visiting Sydney
Observation 1 'Daily Stand up' meeting	20/06/2012	11:00 am	30 mins	Lake project teams	Sydney, China and Melbourne
Observation 2 'Retrospective' meeting	13/11/2012	3:00 pm	1:30 mins	Lake project teams	Sydney, China and Melbourne
Observation 3 'Showcase' meeting	16/11/2012	2:30 pm	1:10 mins	Lake project teams	Sydney, China and Melbourne
Observation 4 'Homemade Jam' meeting	19/11/2012	1:30 pm	40 mins	Lake project teams and all the Ocean Group IT staff	Sydney, China and Melbourne
Observation 5 'Business Analyst' meeting	20/11/2012	2:15 pm	1 hr	Lake project teams	Sydney and China
Observation 6 'Technical Forum' meeting	04/12/2012	3:20 pm	50 mins	Lake project teams	Sydney, China and Melbourne

3.6.4 Data Management

Due to the possibility of analysing a large volume of data, I used management tools for easy analysis. The data for each research question was analysed separately (Yin 2009). Therefore, I stayed organised throughout the process by using certain checklists for all the participants and matrix forms to capture observational field notes (Appendix B) and comments made during the interview (Creswell 2007, Merriam 2009). Also, I used these forms to record thoughts, hunches and speculations that may be integrated with the data (Merriam 2009). Identifying notations were made for each interview, a set of field notes and documents. I developed and used systematic codes consisting of letters and numbers to represent the research subjects for easy retrieval and analysis (Creswell 2007, Merriam 2009). The coded data were organised in data coding tables (Appendix E).

3.7 Data Analysis Approach

I followed a five-phased qualitative data analysis technique including "(1) Compiling, (2) Disassembling, (3) Reassembling (Arraying), (4) Interpreting and (5) Concluding" (Yin 2011, p. 177). This section discusses data analysis phases under the themes of data management and data analysis. Additionally, more details about the case analysis have been provided using suggestions from Creswell (2007), Miles *et al.* (2013) and Stake's (1995) recommendations on data management, reading and memoing, description, classification and interpretation. Chapter 4 details the process and procedures that was used in this study.

3.7.1 Data Analytic Technique: Disassembling, Reassembling & Interpreting

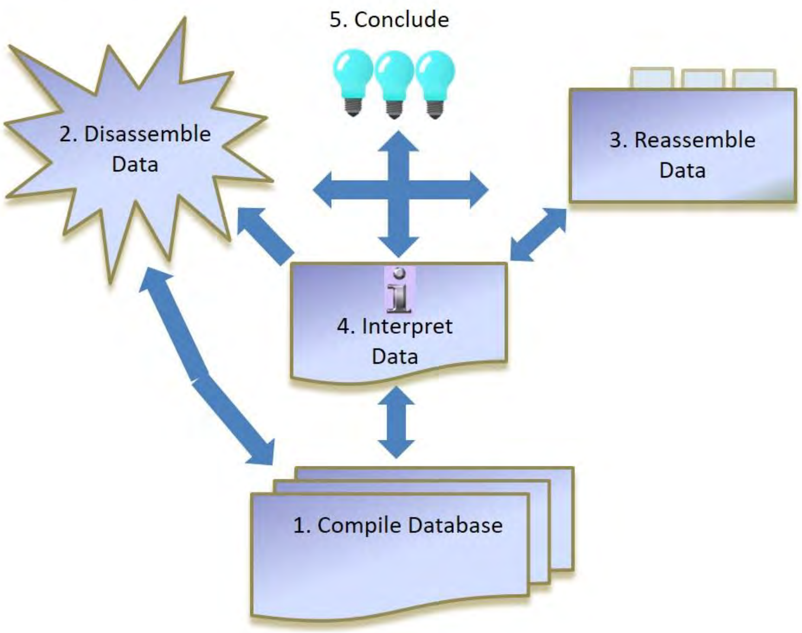
Data analysis and management can be done effectively when one understands how they are defined in qualitative research (Hatch 2002). Data analysis is the transformative process of linking data into research findings (Patton 2002). In other words, data analysis includes "reducing the volume of raw information; sifting trivia from significance; identifying significant patterns; and constructing a framework for communicating the essence of what the data reveals" (Patton 2002, p. 432). The purpose of this transformative process is "to bring meaning, structure, and order to data" (Anfara *et al.* 2002, p. 13). Hence, data analysis can be considered the process of interconnecting data into a narrative form that the reader can understand (Merriam 2002, p. 130). Such linkages could be formulated by identifying common recurring properties or patterns that match the units of analysis (Miles & Huberman 1994).

Technically, data analysis in the qualitative research paradigm coincides with the beginning of data collection (Goulding 2002). Also, data analysis should be situated in "detailed description"; "categorical aggregation" (i.e., results from multiple sources); "direct interpretation" (i.e., results from single instances); "correspondence and patterns" (i.e., matching categories to construct possible patterns or a trend); and development of "naturalistic generalization":

deductions, assertions, and conclusions shaped out of the researcher interactions with the data (Apori-Nkansah 2008, Stake 1995).

As explained in Chapter 3 (Section 3.7), the data analysis was conducted in a number of sequential phases. I used Yin's five phases of data analysis including compiling, disassembling, reassembling, interpreting and conclusion as illustrated in Figure 3.2. The two-way arrows in Figure 3.2 imply that you can go back and forth between two phases. As a result, the entire exhibit suggests how analysis is likely to occur in a nonlinear fashion (Yin 2011).

Figure 3. 2 Data Analysis Phases



Adapted from Yin (2011)

Compiling: Compiling involves reviewing the data immediately after an interview (Yin 2009) to sort the data and put them in order. The aim at this stage is to get a broad understanding of the empirical data (Creswell 2009). In this phase, the researchers tries to be familiarise with its own field notes. In order to do so, the researcher should be continually reviewing the field notes and relistening to recorded interviews to check the recordings repeatedly, again to be familiarise with the data that have collected.

Disassembling: Similarly, disassembling phase contains iterative steps. This phase includes disassembling the data into smaller pieces and assigning codes. The researcher continually go back and forth between the initial ideas about how to disassemble the data and the actual data, potentially leading to modifications to the initial ideas. These kinds of thoughts should themselves be recorded as part of a series of memos kept throughout the analysis (Yin 2011).

In this phase, the researcher begin moving to a slightly higher conceptual level and items that seem to be similar will be assigned the same code. This higher conceptual level will enable the researcher later to sort the items from different records in different ways, such as into similar and dissimilar groups. Once sorted, the researcher can examine the related features of these groups and gain insight into them.

The initial codes are referred to as Level 1 codes or open codes. As the researcher progress in doing this first level of coding, s/he may start to think of ways that some of the Level 1 codes relate to each other, therefore the next goal is to move incrementally to an even higher conceptual level by recognizing the categories within which the Level 1 codes may fall. The coding therefore proceeds to a second and higher set of codes, which can be referred to as Level 2 or category codes (Yin 2011).

Reassembling: This phase is about reassembling and recombining the data using substantive themes or code clusters. This can be achieved by upgrading the level 1 and level 2 coding into a “higher conceptual plane”, where themes and theoretical concepts emerge (Yin, 2011, p. 191). The data can be reassembled by using graphs or arrays in tables (Yin 2011). In this phase, the researcher may become aware of potentially broader patterns in the data and how they might inform the original research questions or how they might reveal some important new insights into the original topic under study. During the reassembling process, the researcher should constantly be querying herself and the data.

Interpreting and Conclusion: Interpreting “the reassembled material to create new narrative” (Yin 2011, p. 179). This phase involves using a wide range of the researcher’s interpretive skills covering the critical and deepest meaning of the data. The objective is to develop a comprehensive interpretation of the collected data. In this last phase, the search for patterns in the coded data focussed on identification of the relationships between constructs as they pertained to the empirical case. The conclusion is discussed in Chapter 7.

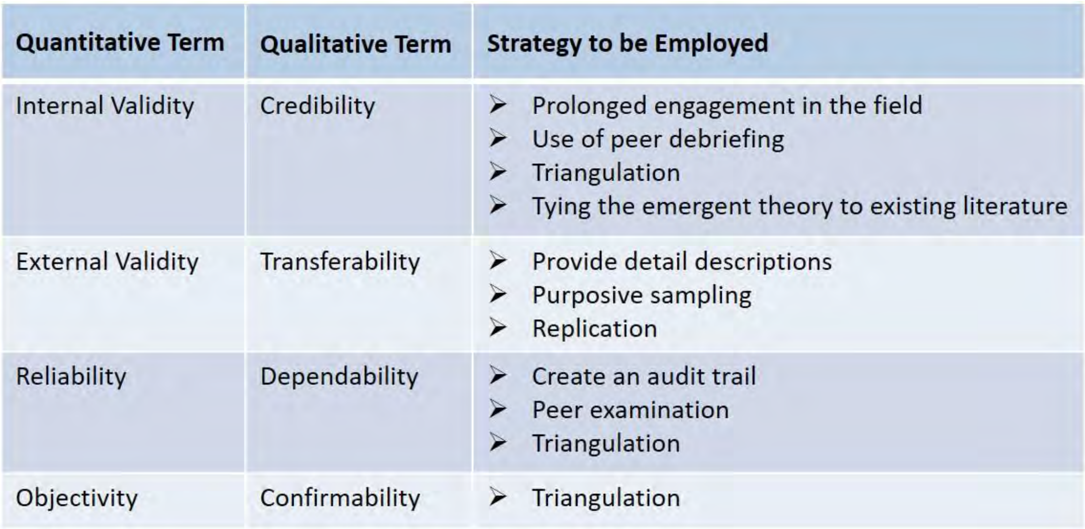
3.8 Issues of Trustworthiness

Validating qualitative results can be done in different ways and still be authenticated. The quality standards for qualitative research differ from quantitative research by definition and procedure (Creswell 2007, Singleton & Straits 2005). For instance, even in the same context of qualitative tradition, the use of reliability and validity differ in meaning to quantitative research (Creswell 2007, McReynolds *et al.* 2001). Instead, reliability and validity are replaced by verbiage such as credibility, trustworthiness and authenticity (Creswell 2007). Verification instead of validity was used by Creswell in order to establish his qualitative research as a distinct methodological approach. Hence, Lincoln (2001) emphasised that validity in qualitative study is not rigid but sort of crystallisation process.

In respect of these assertions, I validated the results of this case study through a process of trustworthiness. Trustworthiness in qualitative research supports that the research purpose findings are worth the attention (Guba & Lincoln 1994). Qualitative researchers such as Lincoln (2001) and Creswell and Miller (2000) have developed validity and reliability criteria based on trustworthiness of the research process. These trustworthiness criteria include many steps that ensure sufficient credibility checks are carried out and the case matches the construction of individuals and groups. In the final analysis, validity is always about truth (Miller 2000).

Conceptual interpretation of the data derived from participants is deemed credible when they are representative of each other (Guba & Lincoln 1994). There are four aspects of trustworthiness in qualitative research, namely; credibility, transferability, dependability and confirmability (Guba & Lincoln 1994, Trochim 2001). Table 3.5 presents the trustworthiness concept that is applied to this research in order to improve the trustworthiness of the study.

Table 3. 5 Criteria for Assessing Quality and Rigor



Quantitative Term	Qualitative Term	Strategy to be Employed
Internal Validity	Credibility	➢ Prolonged engagement in the field ➢ Use of peer debriefing ➢ Triangulation ➢ Tying the emergent theory to existing literature
External Validity	Transferability	➢ Provide detail descriptions ➢ Purposive sampling ➢ Replication
Reliability	Dependability	➢ Create an audit trail ➢ Peer examination ➢ Triangulation
Objectivity	Confirmability	➢ Triangulation

Adopted from Lincoln (2001)

3.8.1 Credibility

Essentially, qualitative case study research shows credibility when it is assessed to be aligned with its initial purpose, which is to report on the perspectives of participants (Guba & Lincoln 1994, Trochim 2001). Therefore, I ensured that the results were consistent with the perspectives of the participants. According to Yin (2009) in all research, consideration must be given to construct validity, credibility, dependability and trustworthiness. The literature states that using multiple sources of evidence is a way to ensure construct validity (Yin 2009). In the current study, I used multiple sources of evidence; interviews, observation, archival records and documents. The specification of the unit of analysis also provides the credibility.

I spent a long period of time collecting data for this research project. I conducted and transcribed interviews. After an initial analysis of the data, I conducted further follow up interviews to get deeper into issues raised and to clarify other issues that had not been adequately explained (Patton 2002). This process helped me to gain depth with the issues under study. Such prolonged engagement in the field allowed me to build rapport with the informants and gain consent to further telephone follow up interviews at a later stage in the research project.

I used peer debriefing within the context of my supervisor questioning the analysis so as to ensure that the account of events represented by the case study was resonating with individuals other than me (Creswell 2000). This was a valuable process in developing the themes and analysing the data in a way that was meaningful and accurate.

When considering what kind of data to collect, a principle to follow is to consider multiple sources of evidence that enhance construct validity (Yin 2003). As Creswell (2007) noted, triangulation of data allows for themes to emerge from different data sources. The use of multiple data collection methods can provide supporting or contradicting evidence between the data sources. Punch (2005) names four sources of data: interviews, observation, participant observation, and documentary data. I used triangulation for this study in the form of collecting data from multiple sources. By using different data collection methods (in-depth semi-structured interviews, documentation and observation) I was able to increase credibility of the results and the findings through confirmation by different data sources (Creswell 2007, Patton 2002 & 2005). Using these data collection methods provided multiple perspectives and data sources which give more credibility to the current research. In addition to the interviews conducted in the form of in-depth semi-structured interviews, I collected documentation about the case study firm from the company website, Intranet and emails. Documentation as a method of data collection was explained previously in section 3.6.2.

Further to looking to the literature to support the findings, I also looked for the findings that were conflicting with the literature and saw this as an opportunity for more creative and deeper insight that could sharpen the limits to generalisability of the current study (Eisenhardt 1989). In order to enhance the credibility, I used the replication logic for theoretically similar results and contrasting results. The making of comparisons was essential to this study in order to stimulate my thinking about different dimensions (Strauss & Corbin 1998).

Finally, another approach that I used to enhance the credibility was through tying the emergent theory to existing literature. Linking results to the literature that discusses similar findings was important because I was able to relate similarities in phenomena that were not associated with each other (Eisenhardt 1989). In this way, this study was able to provide stronger credibility, wider generalisabilit, and higher conceptual level.

3.8.2 Transferability

Transferability refers to an assessment of the extent to which the research findings can be applied beyond the limits of this research (Lincoln & Guba 1985). Qualitative research is deemed dependable by the quality of its elementary processes, namely data collection, data analysis, and theory generation (Patton 2002). The problem of transferability of findings has been one of the most prominent drawbacks of case studies. However, methods such as surveys rely on statistical generalisations; while case studies rely on analytical generalisations (Yin 2003). The purpose of analytical generalisation is to generalise the results of the case study to some broader theory. One method of increasing the transferability of the study is to use replication logic (Figure 3.2). It needs to keep in mind that the generalisability is not the main concern for this study.

I conducted respondent reviews for interview data to make sure any uncovered problem will be resolved to increase the credibility of the research. I provided the detailed description of different situations to illustrate a clear picture for readers. Therefore, I was be able to increase the transferability of the findings.

Purposive sampling intensifies Eisenhardt's (1989) idea of theoretical selection wherein the case is selected because it represents the phenomenon in question. I chose the study case for this research purposefully using strategies outlined by Patton (1990). In addition, rich details of the case study are provided in order to demonstrate transferability.

3.8.3 Dependability

Dependability is demonstrated by triangulating the research findings from the different data sources (Silverman 2000). Creating an audit trail is similar to Yin's (2003) notion of establishing a chain of evidence. For the purpose of the current research study, I created an audit trail. This audit trail allows an external observer to follow the derivation of any evidence reported in this case study (Guba & Lincoln 1994). The audit trail of this study is presented below and followed after an extensive and critical review of the literature:

i. The interview protocol was developed to ensure that all aspects of the research questions are covered

ii. Multiple cases was identified and selected as pilot studies

iii. Access was obtained to pilot cases and the team members participation were secured

iv. I ensured that sufficient resources were accessible and available while in the field such as a tape recorder and a quiet space for the interviews to be conducted

v. In-depth semi-structured interviews with the respondents in the pilot cases were conducted and guided by the interview protocol

vi. I recorded and replayed these interviews in order to refine and revise the interview protocol

vii. When I was satisfied that the protocol covers all aspects of the research questions, a single case was selected for the study

viii. Access was secured and in-depth semi-structured interviews were conducted with the right respondents

ix. I audio recorded and transcribed the interviews

x. The recordings were then replayed to verify accuracy of the transcripts

xi. Documentary evidence on the firm, respondents and the team operation were collected, increasing the construct validity of the study

xii. Following the transcription of each interview and some preliminary analysis, a follow up interview was conducted in order to clarify certain issues and get deeper into others

xiii. I then transcribed the follow up interviews

xiv. A secure database was created for the case study firm wherein the transcripts, audio files, documentary evidence and notes taken during the interview were stored to allow any other investigator access to the documents should the need arise, thereby increasing the trustworthiness of the study

xv. A single firm was selected for the study. The pilot study was not used in the data analysis phase and in the final report

xvi. Single case description is presented later on in next chapter

xvii. The findings are presented after the data analysis phase is completed

xviii. A theoretical discussion of the findings, the contributions and implications are presented and highlighted in the last chapter.

I created a case study database to increase the dependability of the study. This strategy has been highlighted by Yin (2003) as being useful if an external observer were to go back to the case study data in trying to understand the conclusions drawn by the researcher. In the current study the database created and contained all audio files of the interviews, the transcripts and all documentary evidence collected about the case study firm.

3.8.4 Confirmability

Confirmability is the extent to which the data collected supports the research findings (Guba & Lincoln 1994). Researchers are required to document the procedures utilised in order to collaborate and confirm the research findings (Trochim 2001). Consequently, I ensured confirmability by triangulating and documenting all research procedures and findings from the different data sources used in this study.

Triangulation as a means to bring objectivity to the study is recommended by Lincoln (2001) and the triangulation of data in this study is discussed in section 3.6. As a qualitative researcher, I constantly reflected on who s/he is in the inquiry and make his/her biases known as this creates an open and honest account of the data (Creswell 2007).

3.9 Ethical Procedures

According to Tharenou (2000), when conducting social research, all respondents must be treated fairly and ethically. The fair and ethical treatment of respondents in social research involves respecting the personality, rights, beliefs, wishes, consent and freedom of the respondents. With regard to qualitative research, qualitative researchers are guests in the private spaces of the world and as such must adhere to strict codes of ethics (Stake 1995). While conducting this case study research, wherein I had an intense interest in the personal views and circumstances of the respondent, I found it essential to inform the respondent about the disclosure of information gathered and their confidentiality (Stake 1995).

HREA panel introduces a comprehensive list of ethical issues that must be taken into consideration for all research in the university. These issues mainly relate to the content of questions, the way the questions are intended to be distributed and the way that I ask respondents to answer the questions. HREA list is the base for ethical considerations in this proposal and the study strived to satisfy them accurately and honestly. I obtained ethical approval (approval number 116010) from the HREA panel at the University of New South Wales. A detail approach was taken to the letter as it was required by the University of New South Wales ethics approval process. I obtained a letter of cooperation from the Ocean Group as required by HREA. Furthermore, I provided all the case study respondents with:

- The objectives of the research both verbally and in written form
- The contribution to the field that the study intends to make
- How the data would be collected
- What data is required in order to conduct the case study
- When the interviews would be conducted
- When the study will conclude
- What aspects of the information provided the respondents are able to view
- The restricted public access to the case study
- What measures will be adopted through the course of the case study to protect the confidentiality of the respondents

I made clear to the respondents that participation in the study was entirely voluntary and would be terminated at their request should the need arise. Then, I provided a consent form to all seven participants prior to the scheduled interview giving their permission for me to use their answers, in a confidential manner, in this study. I commenced the interview only after the signed consent form was received (Appendix F).

In compliance with HREA and participating organisation guidelines, participants' rights and confidentiality were safeguarded. Throughout this study, I kept the participants’ names confidential to protect the confidentiality of the respondents and prevent their identification. The identification of informants or cases will never be exposed to other informants. They were made aware that I recorded the interviews and then transcribed the recordings verbatim. I did not use any personal information that could be used to identify any individuals.

Protection of the data was another consideration that deserved attention (Goulding 2002, Creswell 2007). I secured all digital recordings, data transcripts, notes, reflections and meeting minutes in a securely locked location that was password protected. The data will be kept for seven years and then destroyed. At all times during the data collection, research methods met the requirements of HREA. Researcher subjectivity was considered and ameliorated by attention to such details. Pure objectivity in qualitative research is not possible because "description, analysis, and interpretation" is required as the researcher moves from "description toward interpretation" (Hatch 2002, p. 9). However, I tried to use their own subjectivities to understand the data gathered from the Lake Project teams. The above measures for participant protection conform to standards suggested by Creswell (2007), Yin (2003), Goulding (2002) and Nachmias and Nachmias (1987).

3.10 Summary

Chapter 3 was an explanation of the qualitative case study methodology chosen to support an examination of the current research questions. This chapter was also a description of the role of the researcher and identified any potential biases and means to alleviate them so they did not taint the narrative. The theoretical method of the research inquiry has been examined in this chapter. The exploratory method is selected for this research, which is to investigate the application of social governance mechanisms in distributed agile development projects in order to understand how to enhance coordination and safeguarding of the project team' exchanges. The distinct concepts of effective governance have emerged in current field research. However, there is little or no detailed in-depth research to shed light on how the social governance mechanisms can be used effectively to enhance the coordination and protect exchanges among distributed agile development project teams.

Using an exploratory single-case study methodology allowed me to construct a rich narrative describing the themes generated from personal interviews of the project team members working together in professional collaboration which is described in Chapter 5. Furthermore, this case study design facilitated the exploration and description of how restricted access, macroculture, collective sanctions and reputation applied in distributed agile development projects and their impact on coordinating and safeguarding the project teams' interactions. It also provided list of effective practices to enhance the coordination and safeguarding of members' exchanges in distributed agile development projects.

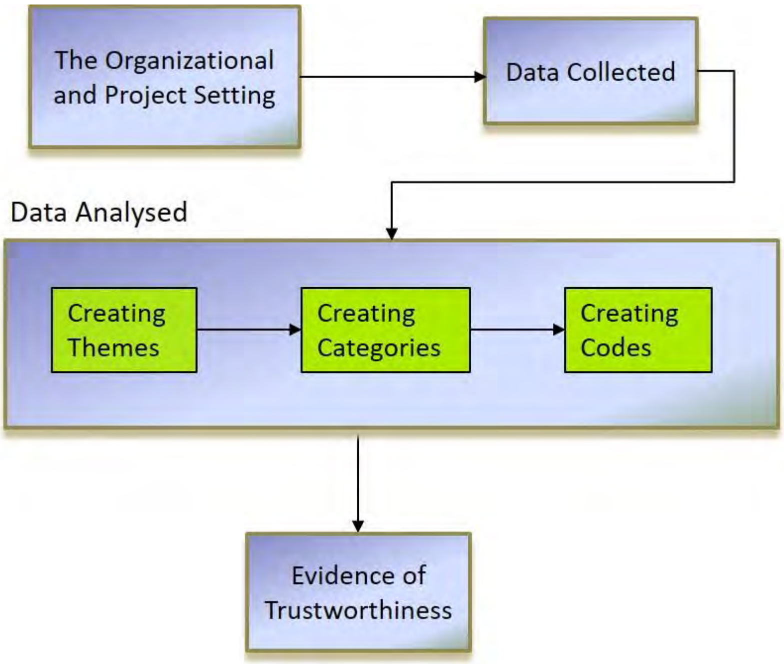
Data was collected through semi-structured interviews, the project artefacts, reports, website data and other forms of information to describe a phenomenological event. Credibility of the data occurred through the triangulation of data and team member checking (Lincoln & Guba 1985). The research design assures the reader of trustworthiness by using an open data analysis and interpretation process. A full discussion of the study's findings from collected data, analysis of data, interpretation based on these findings and implications for further research are in Chapters 5, 6 and 7.

Chapter 4 Data Analysis

4.1 Introduction

Chapter 3 described the research methodology selected for this research and outlined the design, data collection methods and analysis tools used to evaluate the data. The purpose of this chapter is to present the case under study and explains the analysing approach. This chapter begins by providing more details about the project settings (Section 4.2), followed by selection criteria of the Lake Project (Section 4.3) and the details of the data collected (Section 4.4) . Then the study discusses the data analysed (Section 4.5). Finally, Section 4.6 outlines assurance of trustworthiness of these procedures. The study concludes this chapter with a transition to Chapter 5 where the results are explained. Figure 4.1 summarises the structure of this chapter.

Figure 4. 1 The Structure of Chapter 4



Key features of this chapter include specific vignettes related to the successes or disappointments experienced by the teams in the organisation that were conveyed during data collection and answers or comments identifying issues directly related to the research questions.

4.2 The Organisational and Project Setting

This section introduces the case company and project and demonstrates case project selection criteria for this research. For the purpose of this study, a DD project within an Australian company was chosen. The case company is referred to with the pseudonym ‘Ocean Group’. The Ocean Group is an Australian company based in Brisbane, Queensland, with finance, insurance, banking and superannuation business units. The Ocean Group offers services and products to small and medium size businesses, as well as corporate clients. The Ocean Group is a market leader with regard to various services offered to customers. The company’s business strategy is to enhance the value and benefits that come from a single group for clients through simplifying the business and build. The Ocean Group has about 14,500 employees in Australia and New Zealand and relationships with nine million customers. The company has subsidiaries with offices in Queensland, New South Wales, Victoria, ACT, Western Australia, South Australia, Tasmania and Asia-Pacific region.

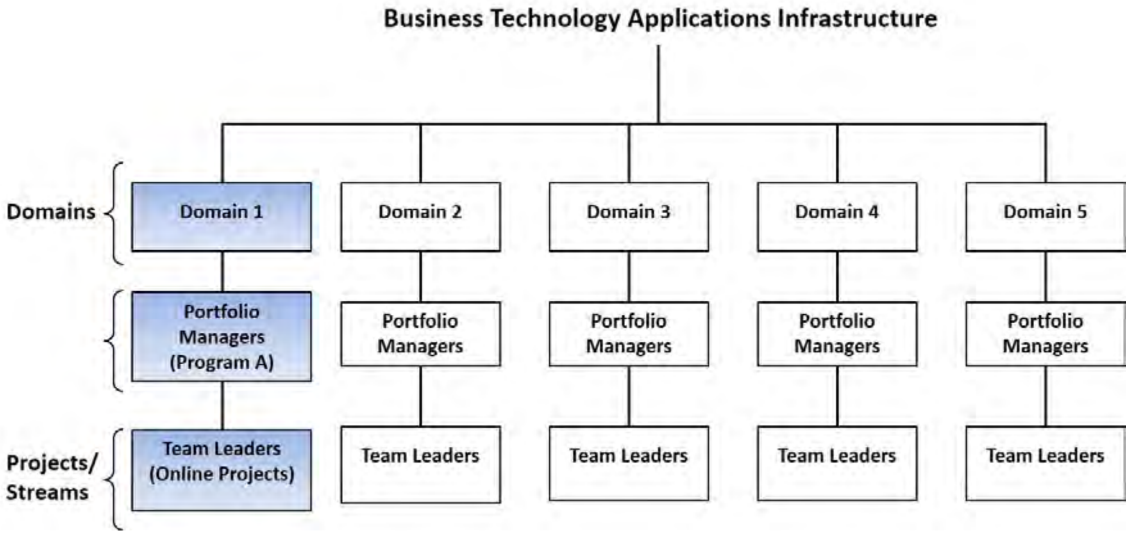
4.2.2 The Ocean Group IT

IT structure for business applications in the Ocean Group are illustrated in Figure 4.2. Enterprise services at a high level are broken down into domains; one of those domains at a program level is accountable for the project. Domains are support for strategic focus for business technology applications in the Ocean Group. Programs rely on all the domains to deliver for them. A program is a collection of projects (streams) that share common business values. In other words, a program is multiple projects where by you might start realising the value as soon as any of the projects are delivered but the business benefit and the entire business case are not realised until all projects in that program are delivered. They are related; projects that combined deliver more value than the individual sum of those projects. The current program under study is program A. Program A has different project managers for different streams of programs. Program A is around moving multiple policy management systems down to one and that’s how business values and savings are done.

Each domain has different programs of work that are managed by portfolio managers. These IT portfolio managers own the relationship with various parts of the business and the team leaders make sure that these parts are delivered. Portfolio managers allocate responsibility of delivery and strategy to each of his/her team leaders based on the work. They work around the business and on all the work that needs to be done. Portfolio managers are accountable for delivery of the project to the business and are online with strategies of the larger corporate structure and strategy. Each portfolio manager may look after more than one portfolio. They can have around 14 projects to manage which would be a very challenging role. It is quite a blanket role; that’s why they have got number of team leaders that look after delivery and strategy for them and monitor the projects more closely (see Table 4.5).

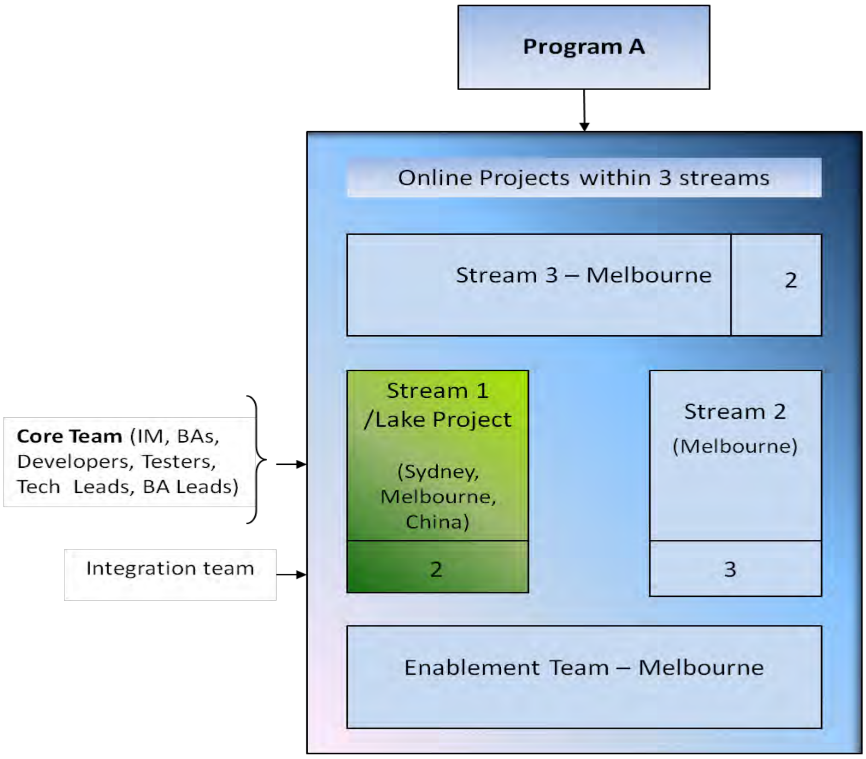
Programs include streams of work (projects). A project in the Ocean Group structure is defined as a set of work effort that needs to be done over a certain amount of time before value can be realised to provide a unique product. The project is carefully planned and designed to achieve a particular goal. The team leaders manage resources for projects. They are responsible for resourcing the team and making sure that as people roll off they then have more people roll on. They build their team based on skills and availability of people within certain domain. They start with IM then start to build the core team around that person. They are also responsible for maintaining the work with external partners. Each team leader is usually responsible for more than one project, depending on the project size. There are some team leaders that have responsibility of 11 teams and usually take care of multiple projects at the same time. They have a hot seat on different floors and they move around as they lead different teams and interact with them directly (see Table 4.5).

Figure 4. 2 The Ocean Group IT HR Structure



Program A includes three streams of work (projects) - one in Sydney and two in Melbourne (see Figure 4.3). Focus of this study is stream1 of this program (the Lake Project) which is distributed across geographic boundaries. From a product perspective, each stream team building product that shares features with other streams' products are placed under one program. For example, the teams in stream1 and stream 2 are building similar products and the team in stream 3 is responsible for building same basic product which sits around stream 1 and stream 2 products, but with some additional features that can be used slightly differently. There is also an enablement team, a team managing all the change required in the system. Besides that, there are two or three people in each stream allocated to build the actual integration and to connect to the application they are building. Stream 1 (Lake Project) has two backend team members in integration the team located in Melbourne. From an IT point of view, these two team members make the integration with the backend so the applications can talk to each other (see Figure 4.3).

Figure 4. 3 Program A configuration


Program A
Online Projects within 3 streams
Stream 3 – Melbourne
2
Stream 1 /Lake Project
(Sydney, Melbourne, China)
2
Stream 2
(Melbourne)
3
Core Team (IM, BAs, Developers, Testers, Tech Leads, BA Leads)
Integration team
Enablement Team – Melbourne

4.2.3 Overview of Ocean Adoption of Agile

The Ocean Group decided to go agile across their entire IT organisation in order to deliver business value faster, cheaper and with lower risk. Within a year of starting the training program, thousands of the Ocean Group IT employees were trained. During this process, the company found out that the crucial areas of leadership, governance, infrastructure and business were not addressed in agile at all. The company also recognised that in order to ensure a successful adaptation of agile within their organisation, they need to provide training to cover the whole spectrum. This meant when the Ocean Group started to apply the agility to its business, they had to extend agile practices to non-IT areas of the organisation as well in order to create the culture of agility for the whole organisation. The non-IT sections' training was specific to their business unit with the aim to provide them with specific coaching and eliminating waste from processes and systems. In this way, agile was introduced to the Ocean Group organisation wide.

The agile methodology is a norm in the whole organisation and used in all projects. All the IT personnel at all company locations are required to apply the company's agile approach. The only negative view that some the project team members have about the agile software development is that there is no standardised or fixed process to follow. It's all about self-managing and empowering the project team members.

The Ocean Group successfully adopted an agile methodology within their organisation through internal training across their entire IT organisation. The training was fully integrated and structured to cover not just the core agile skills and practices in software development life cycle but the entire information management area including leadership, governance, infrastructure and business as well. This agility culture influenced the interactions structure of the organisation, where beside the HR interaction structure (Appendix F - solid lines) you can see informal interactions (Appendix F - dashed lines) between different project team members across the organisation.

Similar to the Ocean Group, the River adopted agile methodology in their company very well. Working agile is part of the way they work and there is no other way of doing the job for the River's staff.

4.2.4 The Lake Project

The case project investigated in this study is referred to with the pseudonym Lake Project. The selection procedure of the Lake Project is discussed in Section 4.3. The Lake Project is the first distributed agile project in the Ocean Group and considered a high profile project. The Lake Project is sourced from the Ocean Group staff including Australian teams distributed in Sydney and Melbourne plus a partner company staff including a China team. The partner company is referred to with the pseudonym 'River', it is an IT consulting company that the Ocean Group pays for their knowledge, experience and services which is a motivation for the River's staff to always behave very professionally. It is the first time that the Ocean Group partnered with the River in China for this project but previously, the Ocean Group used to be partnered with them on ground. The Ocean Group management believes that the River has the best manpower due to its recruitment culture which is very selective. The organisational values and working culture of both companies are very well compatible with each other, which facilitate a similar way of working in both companies.

For the purpose of this study, the focus was on the Lake Project core team. The Lake Project began in January of 2012 as part of a larger program (program A) that had started in Feb of 2011. The Lake Project was an online project with the aim of simplifying online policies. Indeed, the technical strategy of the Ocean Group was moving towards simplifying the policies and legacy system and the Lake Project was one of the projects under the Legacy Simplifying Program (LSP) umbrella. This simplification strategy also influenced the way they developed the projects. For example, the features that are developed in the Lake Project were similar to the ones in team 2 and 3 but instead of writing the codes from scratch, the Lake Project teams share their code base for similar features with the other project teams in the same program. In this way, they tried to simplify the development process.

The Lake Project was considered a high profile project in the Ocean Group organisation. It was the first distributed agile project in the Ocean Group and they took every single effort to make the distributed agility work effectively by end of the project. For example, they chose experienced and senior developers, used different tools to enhance coordination and communication among the Lake Project distributed teams, used mix of agile practices that worked well for the Lake Project, recruited a partner company with similar features for the Lake Project overseas team and implemented co-located distributed teams in the Lake Project through the project.

The Lake Project manager believes that the project was very well established in agile methodology. The agility culture influenced the interactions structure, where in addition to HR interactions among the project teams (Appendix F - solid lines), there were also informal interactions between the teams (Appendix F - dashed lines). For example, colour dashed lines to the Lake Project team (Appendix F) indicates the informal and direct interaction of the project manager with the Lake Project teams. Similarly, the program manager had an indirect and informal interaction with the Lake Project teams instead of having HR interaction that needs to go through the project manager, the program IM then the Lake Project IM (Appendix F).

The Lake Project team leader was responsible for the Lake Project resource management and planning of the core team. The first step was to choose the IM then build the team members around that person. Then the team leader borrowed people from different domains (HR teams) to build the Lake Project delivery team. For example some of the Lake Project core team members were borrowed from domain 3 and domain 4. The Lake Project was a high profile project and special effort was put in to resource the project core team members. The project team leader reviewed the skill sets and selected those individuals who he felt most closely matched the skills needed based on the context and technology selected for the project to deliver the online application. Then people started to move around and were allocated to the Lake Project.

The reporting structure for the Lake Project was different with the other projects/programs within the Ocean Group. Previously when large projects or programs were built, all the core team members were physically moved from their domain and had their delivery reporting structure aligned with their HR reporting structure as well. So they were all reporting to the domain they were coming from. However, with program A, there were two different reporting structures: delivery reporting and HR reporting structure. The difference was due to the fact that the core team members were not physically relocated and the HR reporting lines were maintained. In HR reporting structure, the core team members report to their HR team leader (who is responsible for their annual leave, annual performance, promotions, salary and the like). In the delivery reporting structure, the core team members report to their Lake Project team leader who resourced them for this particular project and is responsible for their performance and delivery for the Lake Project. In other words, from the delivery sense, they actually reporting to the program and from a HR sense they report to the domains.

The delivery team leader interacted with the HR team leader for any of the core team members if required. In this way, the team also had more accountability of the delivery, not just that they had been given the program but they had their reporting structure anywhere they moved. This flexibility enhanced the coordination challenges.

The Lake Project core team members were distributed in three geographical locations; Sydney, Melbourne and China, except the product owner who was based in Queensland. The Sydney team was named as ‘team S’, the Melbourne team as ‘team M’ and the China team as ‘team C’. Team S and team M were assembled by the team leader from available sources in different domains within the Ocean Group but team C was an external team; they were an external company that the Ocean Group was partnering with for duration of the Lake Project.

Team S had 13 team members including four front end developers, one user interface developer, two technical leads, two business analysts (BA), one test lead, one e-commerce representative, one developer and tester and one IM. Team M had five team members including two mainframe developers, one mainframe BA and two testers. Team C had seven team members including four front end developers, one senior BA, one junior BA and one tester. Team S and C had front end Java developers whereas in team M, there were mostly the mainframe developers and mainframe business analysts. The testers in team M were able to work with both mainframe and front end applications (see Appendix E).

Agile software development lifecycle is iterative where high quality adaptive software can be developed by small teams using continuous design improvement and testing based on rapid feedback and change. The development model is to bring rapid value to the business with collaborative management style (Nerur & Balijepally 2007, Boehm 2002). For agile methodology to be successful in practice, it has to be considered as a process together with a series of principles and practices based on the Agile Manifesto (Unhelkar 2016). This applied to the Lake Project as well. The IT personnel at all the Ocean Group locations were required to apply the company's agile approach and the same thing applied to the Lake Project. The Lake Project teams used mix practices from different agile methodologies such as Scrum, Kanban and XP such as distributed pair programming (Section 2.2). The key practices that the Lake Project used in its agile software development approach were (for details on agile practices see for example Dyba & Dingsoyr (2008), Abrahamsson *et al.* (2002):

I. use of an electronic story wall (JIRA) and physical story wall
II. continuous incremental testing; day to day testing by testers and sometimes by BAs or developers
III. IPMs (Iteration Planning Meetings); every Monday at the beginning of the iteration
IV. iteration development cycles; every two weeks a team develops a working and tested software increment that can be demonstrated to the business representative and released for user acceptance testing
V. daily Stand ups; sometimes product owner participated (Section 3.6.3)

VI. daily Scrum of Scrums; IM participated in this daily meeting as the Lake Project representative to discuss other project (stream) reporting completion, the next steps and coordination challenges between the teams

VII. Retrospectives; every two weeks and every stakeholder was involved (Section 3.6.3)

VIII. pair programming; co-located or remotely through messaging system

IX. Showcase meetings; every two weeks to all the project team members including the core team, sponsors, BAs, PO and E-commerce (Section 3.6.3)

X. Developers' (Technical) Forum: (Section 3.6.3)

XI. BAs' Forum: (Section 3.6.3)

XII. Testers' Forum; similar to other forums where testers across locations meet and interact to discuss completed tasks, pending tasks and current issues

XIII. Tech Lead Forum; tech leads across three streams of 1, 2 and 3 get together. The forum is set up for making decisions on, for example, technology problems they need to resolve

XIV. Tech Huddle: after the daily Stand up, developers get together and talk about what they are doing. It's another form of Stand up but it's more about what are they working on today because the application becomes so big and a developer working in isolation doesn't know what's going on. Developers in each stream have a tech huddle, so for team 1, they have this all in China, Sydney and Melbourne together

XV. IPM; an Iteration Planning Meeting to discuss what the team wants to achieve in next iteration (Section 3.6.3)

The Lake Project was a high profile distributed project with two reporting structures: delivery and HR reporting structures. The Lake Project was not only distributed across locations but it was very well established in agile methodology and the Lake Project teams used some of the key agile practices as part of its agile software development approach. These specifications made the project an eligible project case for the purpose of this research.

4.3 Selection Criteria of the Lake Project

As explained in section 3.4, I followed the six criteria proposed by Miles and Huberman (1994) to evaluate the sampling and selection of the case for this research. The Lake Project was selected because firstly, it was well distributed across geographic boundaries. The Lake Project teams were distributed across four geographical locations, Sydney (NSW), Melbourne (VIC), Chengdu (China) and Brisbane (Queensland). Secondly, based on NG theory, the Ocean Group can be considered a network because it consists of independent subsidiaries with offices at various locations. Moreover, while the majority of the team members are allocated to one team from one subsidiary only, when required they will collaborate across teams and locations. In addition, in order to be able to apply NG theory model, the case required to meet the four preconditions for the NG theory (Table 2.5). The applicability of these preconditions in the project case is explained in detail below.

The first precondition for NG concerns demand uncertainty. NG is found in industries with high levels of demand uncertainty but a relatively stable supply. In addition, demand uncertainty is generated by unknown and rapid shifts in consumer preferences (Jones *et al*. 1997). Information technology industry meets these criteria very well as does the selected project case. Demand uncertainty is also generated by rapid changes in knowledge, or technology which results in short product lifecycles (Powell 1990). In high technology industries such as information technology, new products and technologies leap prior products and technologies, leaving participants scrambling to catch up or get left behind (Jones *et al.* 1997). This is very much compatible with the selected project in the competitive IT industry. The Lake Project product was an online application characterised by a high level of demand uncertainty due to fierce competition and rapid changes in technology and user expectations and requirements. For the agile teams, the demand uncertainty manifests itself as changes in the business's requirements and priorities. To respond to the demand uncertainty, the team aims to get new versions of the software products to market as soon as possible.

The second precondition concerns the execution of complex tasks under time pressure. Time pressures are due to the need to reduce lead time in rapidly changing markets such as information technology (Jones *et al.* 1997). The selected software development projects integrate multiple autonomous, diversely skilled team members under intense time pressures to create complex software products. Such a need for speeding products and services to market is a critical condition for networks (Powel 1990). The development of the software can also be considered as series of complex tasks executed under time pressure because the company has a two-week release cycle. The task complexity is increased further because business and IT people continually enhance the functionality and adapt the capabilities of the product platforms.

The third precondition relates to customised asset-specific exchanges. Customised exchanges with high levels of human-asset specificity require a structure that enhances cooperation, proximity and frequent exchanges to effectively transfer tacit knowledge among the teams (Jones *et al.* 1997). Cooperation among exchange members is necessary because teams must work together to gain tacit knowledge (Jones *et al.* 1997). In the Lake Project the interactions between the business and IT people are characterised by a high degree of human asset specificity. This could be because both have to develop in-depth knowledge of each other's domains and goals to be able to co-produce a satisfactory end product. The project's structure also facilitates the cooperation among the project teams. At the same time, the exchange outcome is a fully customised product platform based on in-house developed back-ends and software engines rather than standard software components.

The fourth precondition is that of frequent exchanges between the project teams. Frequency concerns how often specific parties exchange information (Jones *et al.* 1997). These settings and structures do exist in the Lake Project because the project teams, although distributed across geographic boundaries, have frequent interactions and exchanges among themselves on daily basis. This is facilitated through effective communication tools which results in the transfer of tacit knowledge. In addition, to successfully co-produce the customised software product, frequent interactions between the business representative and the product manager as well as within the team are necessary. These interactions take place in planned meetings, such as iteration planning (on a two-weekly basis) and review meetings, daily Stand ups and Retrospectives, as well as via other practices specified by the applied agile method, such as pair programming. In addition, much face-to-face communication occurs to create and respond to change.

4.4 Data collected

So far, the questions of what this study intends to find out, why, and from whom have been addressed. The next issue is to describe how the data has been collected. This section is going to explain in detail how the data were collected for this study. In preparation for collecting the project related data and to answer the research questions, as was explained in Chapter 3 (Section 3.8.1) I noted the importance of triangulating data. Therefore, I used variety of data gathering techniques including documents, observation, in-depth semi-structured interviews and attendance at meetings (see Table 4.1). Data collection began after I had obtained HREA approval (Approval Number: 116010) to conduct the research.

In line with the explorative aim of this research, initially an interview protocol for the pilot study was used that contained open-ended questions about broad themes. The following themes were included: (1) general information about the company; (2) details on the agile software development practices used; (3) positive and negative implications of using agile software development; and (4) interpersonal aspects of the agile software development. The interpersonal aspects of relevance for this research relate to the use of social and interaction mechanisms and their impacts on coordination and safeguarding exchanges among the project teams across locations. This protocol was a useful tool in pilot studies to test the questions, timing, and interviews' answers. After practicing this protocol in pilot studies, I was able to write more precise protocol and questions to cover the intended subject fully.

The actual interview protocol that used for the case under study is explained in Appendix D. All the interviews followed the same format and the questions in Appendix D were asked as standard. At the beginning of each interviews, I went through some introductory remarks, then I went through the standard questions for each constructs. There were times I have sought some follow up questions. Supplementary questions were asked in terms of clarification or seeking further details of an item raised. At the end of each interview, there was time for questions or to add anything that was necessary. The interviews ran in sequence however, sufficient questions were asked in and around each of the main constructs to provide sufficient detail. I developed the interview protocol and added questions on the basis that the interviewees were typically answered the questions without being asked. The protocol was to be used to initiate the questions and it was also a check list to make sure that I have covered the topics required. As expected, the interviews mostly stayed within the structure in the protocol however, once the discussion on a particular construct was initiated and the first question on each construct was asked, the interviewee provided most of the information being sought without further prompting. So, I would let the interviewees speak and as they had mentioned the answer to the subsequent questions I would cross them off. Then, as the section was drawn to the end, if I had any uncrossed questions, I would then asked the interviewees at that point.

Data collection took place from June to December 2012. In this time period I visited the company site 18 times (Table 3.3) and made observations, held interviews, gathered company and the project documents and wrote field notes (Appendix B). The interviews constitute the primary data source, while the gathered documents and field notes serve as background material that allow for further contextual understanding.

I collected data through number of artefacts from the case company and I acquired documents from the company listed in Table 4.1 Data was collected from semi-structured interviews in the form of recorded audio captured during each interview. The audio files were copied from the recording device and uploaded to my personal laptop where I had the ability to listen to each of the recordings. At this point I reviewed each recorded interview, transcribed them and added to existing notes some of the key areas of interest that emerged from the recorded conversations. The notes, along with a review of the transcripts, were used to develop codes from which categorisation and thematic analysis could begin (Merriam, 2009).

Table 4. 1 The Project Artefacts

Document Type	Document Name	Context	Where listed
Project artefacts	The project structure chart	The way a the case project and teams are structured	Appendix J
	Ocean Group values statement	The values that Ocean Group follows	Appendix H (item 1)
	River values statement	The values that River follows	Appendix H (item 2)
	Lake project relational contract document	List of points that the Lake project team members agreed to follow	Appendix H (item 3)
	IT HR structure chart	Two ways of HR and IT reporting system	Figure 4.2
	Program structure chart	The way the program A is structured	Figure 4.3
	Project status report	The project status including budget report, risk management report and issues report.	Appendix H (item 4)
	Project team activity report	The activities and progress of the team members	Appendix H (item 5)
	Retrospective plan	The details of what is required to be discussed	Appendix H (item 6)
	The team meeting minutes	The details of the current issues	Appendix H (item 7)
	Project Performance report	The analysis of the project performance	Appendix H (item 8)
Website data	The case company web site	Background about the case company The company's structure The company's business strategy	Information obtained used in section 4.2.2, 4.2.3 and 4.2.4
Interviews	Semi-structure interviews		Appendix D

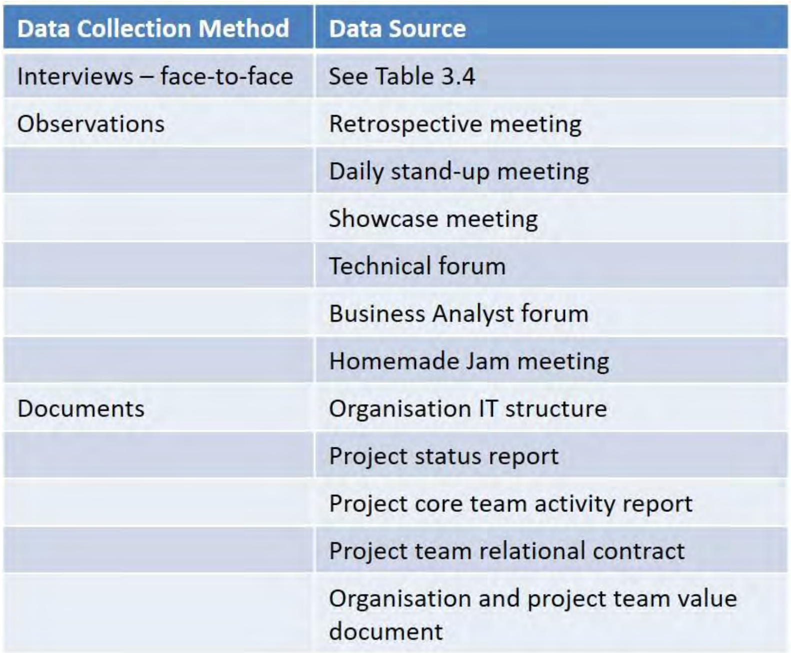
Discovered themes were used to help the research "see" (Boyatzis 1998, p. 4) and understand the application of social governance mechanisms in distributed agile development projects. As data from the recorded audio was analysed and themes uncovered, I cross-referenced the information with data collected from the selected project. While most of the data corresponded with one another, one point of clarification required additional queries to some of the interview participants. This constant comparison of the information seen in the data (Charmaz 2006), coupled with reflection about each related topic as outlined in Chapter 2, allowed me to understand how each of the social governance mechanisms were applied in the project, what were the challenges and effective practices to face these challenges and how these mechanisms enhanced the coordination and safeguarding of the project teams interactions across geographic boundaries.

A total of seven semi-structured face-to-face interviews were conducted within 7 main interviews, 3 continuation interviews and 2 follow up interviews. The participants were management and different core project team members (see Table 4.1). I chose the interviewees because these were the people I identified that I needed to talk to (Section 3.3). Appendix I provides detailed information of the interview participants. Each interview was generally of one to two hour duration and they were audio-recorded with permission to ensure accuracy of the information gathered from each participant. Notes were taken during each interview as another form of documentation, and to provide a backup to the audio-recorded conversation in the event of a malfunction of the recording device. Interviews were complemented by comprehensive reviews of documents and discussions at follow up interviews. The nature of the follow up interviews were to clarify number of points and they have no material bearing on the code. These information were to simply add further depths to the information already gathered but they did not provide specific additional information. Interview 2, 3 and 5 had to be rescheduled and continued on because of time constraints from the interviewee (Table 3.4).

The follow up emails and telephone conversations were also used to clarify and refine issues that emerged during transcription (Corbin & Strauss 2008). All interviews were subsequently transcribed, converted from the tape record to text, which resulted in more than 275 pages of interview transcripts.

For the semi-structured interviews, open-ended questions were used to generate discussion and gain insight into the experiences of the project team members. Since this research is focussed on adapting effective social governance mechanisms in distributed agile development projects, it was important to begin with questions in a semi-structured format. Allowing for discussion around each question led to qualitative information that helped to answer the primary research questions. It also provided an avenue in which additional questions could be asked to provide clarification to responses. It provided insight into how social governance mechanisms are applied and their impact on coordination and safeguarding of the project teams' interactions.

Table 4. 2 Data Source for the Qualitative Study



Data Collection Method	Data Source
Interviews – face-to-face	See Table 3.4
Observations	Retrospective meeting
	Daily stand-up meeting
	Showcase meeting
	Technical forum
	Business Analyst forum
	Homemade Jam meeting
Documents	Organisation IT structure
	Project status report
	Project core team activity report
	Project team relational contract
	Organisation and project team value document

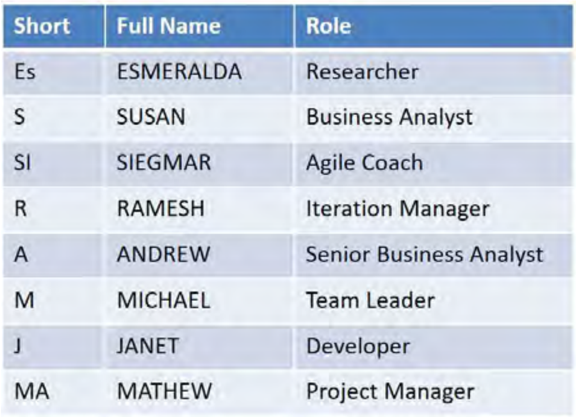
During the first four months, project data, reports, website data and other information were collected. The last two months was used for additional interviews for clarification on information gathered during the first months and to give interviewees an opportunity to review and validate information provided by them, along with my interpretations. In total, I conducted 12 interviews and participated in six observation sessions. During the period of data collection, I diligently followed the data collection sequence described in Chapter 3. I recorded all the interviews and maintained the protocol described in Chapter 3 for minimising risks and protecting participants. Participants were given a letter of appreciation upon concluding the research (see Appendix A). There were no variations from the data collection plan outlined in Chapter 3 nor were there any unusual circumstances to report.

4.5 Data Analysed

As explained in Chapter 3 (Section 3.7), the data analysis was conducted in a number of sequential phases. I used Yin’s five phases of data analysis including compiling, disassembling, reassembling, interpreting and conclusion (Yin 2011), as follows:

Compiling: In this research, I reviewed the empirical data (i.e. the interview, collected documents and field notes) several times and performed a review after each interview to make notes about the interview, and followed up for clarity on any recording or notation that was not clear. No coding took place, but notes and impressions were written down, reviewed frequently and then sorted. I downloaded the interview recordings and saved on a password protected file on my computer. Each file interview was saved on my removable hard drive which was password protected. Table 4.3 presents the research participants and how I refer to them in coding section as the code source.

Table 4. 3 Research Participants



Short	Full Name	Role
Es	ESMERALDA	Researcher
S	SUSAN	Business Analyst
SI	SIEGMAR	Agile Coach
R	RAMESH	Iteration Manager
A	ANDREW	Senior Business Analyst
M	MICHAEL	Team Leader
J	JANET	Developer
MA	MATHEW	Project Manager

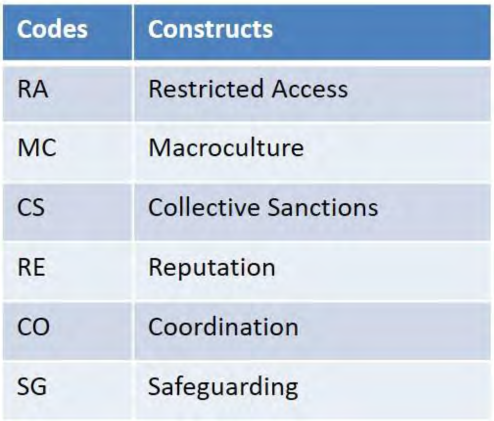
Disassembling: identifying constructs

The second phase of data analysis, which is recursive, required reviewing the data and breaking them down into smaller pieces (Yin 2009). In order to attain understanding of the data collected, I tried to make sense of the data through a continuous process of "reading and memoing" (Creswell 2007, p. 143). Memoing means to note "short phrases, ideas, or key concepts that occur to the reader" (Creswell 2007, p. 151). At the stage of reading and memoing, I read the entire material several times and compared the data with my initial ideas about disassembling the data (Miles et al. 2013). The materials to be disassembled consisted of documents, interviews and field notes.

To disassemble the data, I converted the interview data from tape to text and wrote the transcript for all the interviews in a single document and I then looked for each of the constructs in the transcript. Following Miles & Huberman (1994), a more systematic analysis was conducted. Data was analysed using the principles of the template analysis technique (Miles & Huberman 1994). Using the template approach the data was analysed through what is known as a “codebook” (Crabtree & Miller 1992) or a “start list” (Miles & Huberman 1994). A provisional start list of codes/themes was created (Table 4.4) and the list was developed from the list of the research questions, problem areas and key variables that I brought to the study (Miles & Huberman, 1994). I used the NG model as a theoretical lens to create the coding scheme. The use of a-priori specified and defined theoretical constructs and related codes helped to shape the analysis. It also facilitated the creation of a logical link between the theoretical model, the codes and the empirical findings (Dubé & Paré 2003, Eisenhardt 1989).

I identified main themes within the interviews. Using codes from coding constructs level 1, I went through the transcript and read the interviews, then blocked and categorised the comments for each of the different sections which were related to one of the six themes of RA, MC, CS, RE, CO and SG. Each of the interviews were coded across the six level 1 constructs and that was done for all the interviews. Given the nature of the discussion and that the structure of the interviews were about the six different constructs, for the most parts, it was fairly straight forward. I was looking for what the person was talking about, by looking at the words that was being said I could see to which construct it related. The comments were clearly related to one or other construct, very few ever related to more than one construct and no additional constructs was found. I retained the source of the codes at all time.

Table 4. 4 Constructs Codes List



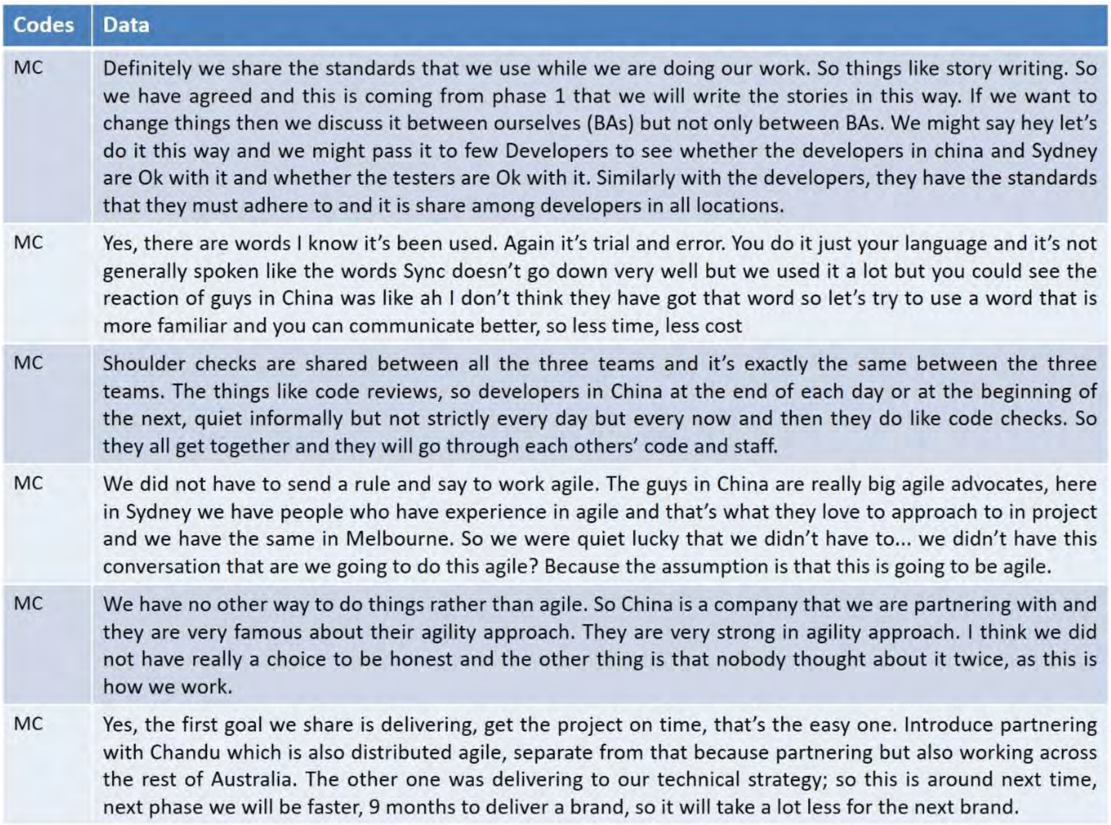
Codes	Constructs
RA	Restricted Access
MC	Macroculture
CS	Collective Sanctions
RE	Reputation
CO	Coordination
SG	Safeguarding

Once level 1 coding had been done, then I collated all of the similar codes across all of the interviews into a similar document for each code (code book 1). I used tables to categorise similar topics that may help to examine certain types of data (Table 4.5). Once all of the interviews were done, I then collated all of the things that I've done for each construct such as MC (Table 4.5), across all the interviews into a separate document (Code book 1) making sure that I have done the main steps. I did that for all other constructs as well. Code book 1 is a lengthy document and Table 4.5 is an excerpt of codebook 1 related to MC.

Then at the end of coding, I reviewed the document again to make sure that my coding was done properly. Each section I looked at, I asked myself to what construct does this relate. Some of them were very evident, those that weren't, I looked at the context in which was being discussed and then reviewed the comment.

In doing the coding, although I was aware that each section should code for a code but I made myself mindful of that there might be other constructs or other codes and did not force the code. So, where I was unsure of the code, I left it uncoded and marked it with question mark for review. For the level 1 coding on the transcripts, on the final review I looked for sections that were not coded and questioned them to see if that meant anything. Some parts of the uncoded sections were not coded for anything and other parts were coded after couple of review iterations. As part of data analysis phase, the findings with case study participants were checked to enhance the validity of data.

Table 4. 5 Macroculture Coding - Level 1 (Code book 1)



Codes	Data
MC	Definitely we share the standards that we use while we are doing our work. So things like story writing. So we have agreed and this is coming from phase 1 that we will write the stories in this way. If we want to change things then we discuss it between ourselves (BAs) but not only between BAs. We might say hey let's do it this way and we might pass it to few Developers to see whether the developers in china and Sydney are Ok with it and whether the testers are Ok with it. Similarly with the developers, they have the standards that they must adhere to and it is share among developers in all locations.
MC	Yes, there are words I know it's been used. Again it's trial and error. You do it just your language and it's not generally spoken like the words Sync doesn't go down very well but we used it a lot but you could see the reaction of guys in China was like ah I don't think they have got that word so let's try to use a word that is more familiar and you can communicate better, so less time, less cost
MC	Shoulder checks are shared between all the three teams and it's exactly the same between the three teams. The things like code reviews, so developers in China at the end of each day or at the beginning of the next, quiet informally but not strictly every day but every now and then they do like code checks. So they all get together and they will go through each others' code and staff.
MC	We did not have to send a rule and say to work agile. The guys in China are really big agile advocates, here in Sydney we have people who have experience in agile and that's what they love to approach to in project and we have the same in Melbourne. So we were quiet lucky that we didn't have to... we didn't have this conversation that are we going to do this agile? Because the assumption is that this is going to be agile.
MC	We have no other way to do things rather than agile. So China is a company that we are partnering with and they are very famous about their agility approach. They are very strong in agility approach. I think we did not have really a choice to be honest and the other thing is that nobody thought about it twice, as this is how we work.
MC	Yes, the first goal we share is delivering, get the project on time, that's the easy one. Introduce partnering with Chandu which is also distributed agile, separate from that because partnering but also working across the rest of Australia. The other one was delivering to our technical strategy; so this is around next time, next phase we will be faster, 9 months to deliver a brand, so it will take a lot less for the next brand.

Reassembling: identifying indicators

In a subsequent data analysis phase, in looking for patterns, data were recombined and themes and categories emerged where the level 2 coding formed. For level 2 coding, I went through the code book 1 and coded that down into a lower level (level 2). I was looking for particular themes, I coded it code by code. I reviewed the whole transcript document for each single indicator. For example, I picked RA1 (embeddedness) and I went through all of the items that I had and those which were related to embeddedness. For RA codes, I went through the collated list 8 times. When all of RA indicators were done and then all of MC, CS, RE, CO and SG were done.

Since this phase involves a recursive process, I applied some aspects of the second phase to the level 1 coding and categories to generate level 2. I used tables (Appendix E) to rearrange the data and extracted the data and themes (Miles *et al.* 2013) by reviewing the frequency of key phrases. The empirical indicators for each of the theoretical constructs were compiled. The empirical indicators delineate the meaning and manifestation of the four social mechanisms as well as of coordination and safeguarding in the case under study.

In level 2, all codes related to RA, CS, RE, CO and SG coded relatively straight forward and there was key distinction between each of the different indicators. However, in Macroculture, the two codes, MC1 (shared norms) and MC4 (shared assumptions), was often difficult to distinguish between them and there was some confusion. When I was coding, I reviewed all the transcripts and these codes and the other codes that looked for two, I coded for both and marked them to be reviewed. Then I came back and reviewed all of the ones that were dual coded. I then questioned myself if I am looking at the third thing or I am looking at something that is legitimately both. I pulled those codes out and look at them in isolation, I reviewed the definitions of the codes and I then satisfied myself that what they actually talked about were two separate themes. Appendix E sets out level 2 coding tables (Code book 2) of the six constructs and their indicators and Appendix F presents quotes for each constructs and indicators which is an excerpt of codeboo2 for each indicator.

Interpreting and Conclusion: The fourth stage consisted of interpreting the data based on Yin's (2011) five attributes which are "(a) completeness, (b) fairness, (c) empirical accuracy, (d) value-added, and (e) credibility" (p. 207). Therefore, all the transcripts were analysed and the ensuing interpretations accounted for all the data available. At the end of the third stage, I followed up with some of the respondents for their feedback on the facts generated because respondents could only confirm or disconfirm fact but could not change the interpretation of the facts (Yin 2011). I used a storyline memo (Appendix M) to capture emerging understanding of how the four social mechanisms facilitate coordination and safeguarding of exchanges in the Lake Project (Creswell 2009). Finally an examination of the coded data for conceptual patterns and linkages enabled me to uncover underlying meanings of the words of the content and see broader themes. That is how I was able to come up with the additional noteworthy points for the study. At this last stage of the analysis, I concluded and explained the significance of the entire case study, which included the academic and practical implications of this case study. The last stage, conclusion, will be discussed in Chapter 7.

Ideological Similarities: In level 2 coding, all of the RA codes were straightforward but after I collated all of the RA codes together, I then coded those and then from that I found that there were sections that were not coded and those ones were the new code. A table created for all the uncoded sections.

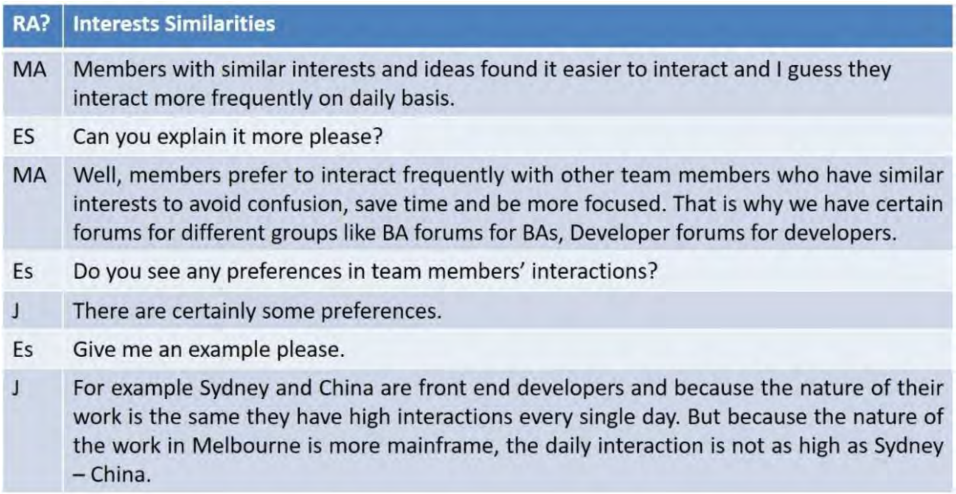
Once I finished the coding, I went through the transcript again and marked and addressed all of the uncoded sections. All the sections that had not been coded, I then separately collated them and then I asked myself are these all about the same thing. After reviewing the uncoded sections, I found out that the parts that were not coded were all related. I looked for specific words and I identified that they talked about similarities among the members that was related to their interests. I found the word 'similarities' in all of them. So I thought to myself is this the code? Is that what they are talking about? I then said, will all the other sections code for that? The ideological similarities coding tables were created for all of the uncoded sections. The first question I asked myself was, are they all about the same thing or are they about different things? After couple of reviews I found that they all are talking about the same thing. I then reviewed to look for specific words. I then re-reviewed all of the RAs to find out if I have missed other instances which might be related to this section.

The key construct that was emerged was ideological similarities. Following the method that I described above, into how I looked for codes, I came up with this construct. I initially code it as 'RA?' and I initially called it 'Interests Similarities'. I went through the original transcript and reviewed the whole document again. It took 6 iterations to fall on this notion of 'ideological similarities' and code it as RA5. The following tables are the initial coding tables. After it was coded as RA5, the code was added to codebook 2 as part of RA indicators (Appendices E and F).

Table 4. 6 The 'ideological similarities' Initial Coding Tables

RA?	Interests Similarities
R	So, it's like daily people from same discipline catch up throughout the day…So it's kind of stand up but only for specific discipline then we've got BA catch up.
ES	Why did you arranged such short catch ups?
R	We found it is easier for people with similar interests like BAs and Developers to interact.
ES	Do you mean there is a preference in their interactions?
R	There is certainly a preference between members to communicate with other members that have similar interests. I guess that is how we came up with all the different forums for them.
ES	What advantages do you get out of these catch ups?
R	The interaction between these members are faster and easier. We actually call them forums like BA forums. It's easier for people in these forums to interact.
ES	Do you prefer or have more frequent interactions with certain members across locations?
S	More with BAs I would say because it's easier and it's probably more necessary. I guess it could be because we share similar interests as well. The BAs spend a lot of time communicating across geographic boundaries.
ES	Can you give me an example please?
S	…especially in China when there is complex subject that needs to be communicated, team members preferred to go to their geographical representatives [such as BA] and asking them to explain, give context and let them to re-communicate it in Chinese for them.

RA?	Interests Similarities
A	…sometimes the interactions are limited to fewer members.
ES	Give me an example.
A	For example in BA forums we only interact with other BAs. It makes the interaction to be easier and more focused. In the forum we share similar opinions and interests. When we have fewer members involved and the interaction between us is more focused we get much better result. It is the same case for other forums like Tech forums and Dev forums.
ES	How do you benefit from these forums?
A	It creates a strong channel among BAs within the team.
ES	What do you mean?
A	I mean having this level of contact between BAs is required otherwise you kind of having a lot of people who…I mean even this as a distributed team, especially the large team like we have, the BA spend a lot of time on fragmented tasks. So rather than sitting and doing, saying I am going to block out one hour to do this particular task, that you constantly have half a dozen of things that need to be done. If we didn't have that really strong channel between the BAs, everything becomes so fragmented and no one has the visibility about what anyone else is doing or what has been done or what changed are made.
ES	How does these forums help interactions among team members?
A	I would say, team members prefer to interact with those members they find having similar interests. For example, as a BA, I prefer to have frequent interaction with another BA that is following similar approach as mine because it is simply easier.



RA?	Interests Similarities
MA	Members with similar interests and ideas found it easier to interact and I guess they interact more frequently on daily basis.
ES	Can you explain it more please?
MA	Well, members prefer to interact frequently with other team members who have similar interests to avoid confusion, save time and be more focused. That is why we have certain forums for different groups like BA forums for BAs, Developer forums for developers.
Es	Do you see any preferences in team members' interactions?
J	There are certainly some preferences.
Es	Give me an example please.
J	For example Sydney and China are front end developers and because the nature of their work is the same they have high interactions every single day. But because the nature of the work in Melbourne is more mainframe, the daily interaction is not as high as Sydney – China.

4.6 Evidence of Trustworthiness

The issue of reliability and validity is well established in quantitative research, however discussion on reliability and validity is often the “exception rather than the rule” with qualitative researchers (Andersen & Skaates 2004). Researchers such as Yin (2003), Eisenhardt (1989) and Miles and Huberman (1994) argue that studies need more than the researcher’s “claim of relevance” while assessing the value of the study’s findings and conclusions.

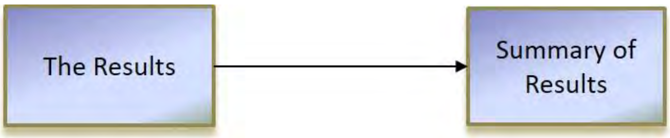
Trustworthiness, authenticity and quality of the research have been addressed as follows (Golafshani 2003). First, to improve the authenticity of the research, an ‘audit trail’ that documents the research process from data collection through to the drawing of conclusions has been produced. Second, to ensure the trustworthiness of my research findings, rigorous data analysis procedures was applied as recommended by extant literature in the field of qualitative research (see for example; Miles & Huberman 1994, Creswell 2009, Lincoln & Guba 1985). Lastly, ‘peer debriefing’ was used (Creswell 2009). The peer was able to apply an outsider’s perspective to assess the plausibility of the empirical findings and to help enhance the readability of the case study accounts. Next chapter discusses the result of the case study.

Chapter 5 Results

5.1 Introduction

The purpose of this chapter is to present the answers to the questions guiding this case study. . This section explains the application of four social mechanisms and how they facilitate coordination and safeguarding of exchanges. Section 5.1 presents the findings from themes that emerged during data analysis. Subsequently, a summary of findings is discussed in Section 5.6. Finally, the study concludes this chapter with a transition to Chapter 6 where the results are reviewed in relation to the literature analysed in Chapter 2. Tables 5.1 5.2 and 5.3 summarise the results. Figure 5.1 summarises the structure of this chapter.

Figure 5. 1 The Structure of Chapter 5



5.2 Research Question 1

The first research question is "*How are social governance mechanisms applied in distributed agile development projects?"* The question investigated the application of social governance mechanisms in distributed agile development projects. To address this question this study conducted an in-depth investigation into the Lake Project and found out how restricted access, macroculture, collective sanction and reputation are applied in the Lake Project teams. The following table summarises the findings:

Table 5. 1 How are Social Governance Mechanisms Applied in Distributed Agile Development Projects?

Social Mechanisms (Jones et al. 1997)	Constructs Confirmed (Jones et al. 1997)	Application in the Case
Restricted access	Status Maximisation	Similar status members interactions more frequently
	Embeddedness	Fewer members interact more frequently
	Ideological Similarities (New construct found)	Members with similar interests interacts more frequently
	Relational Contracting	Membership motivates fewer member to interact
Macroculture	Shared norms	Shared common agile practices Shared non-agile practices Shared common working standards Shared common strategies Shared common approaches Shared common language
	Shared values	Ocean Group values River values Agile values Relational contract values
	Shared goals	Shared goal of delivering the project on time Shared goal of delivering to the project technical strategy Shared focus on delivering software to the satisfaction of the business Shared goal of making the distributed agile component work
	Shared assumptions	To communicate the same message to members and the business To work agile Not to work on weekends and until late hours Either participate in all team activities or challenge them but cannot withdraw from them To discuss issues comes up during iteration with lead BA or IM To monitor blockers on the wall
Collective sanctions	Actions to condemn the unacceptable behaviour	Financial consequences (i.e. losing bonus)
		Non-financial consequences (i.e. assign different task, gradual escalation, spreading words for misbehaviour)
	Actions to reward the acceptable behaviour (New construct found)	Formal rewarding actions (i.e. financial bonus , Thank You System)
		Informal rewarding actions (i.e. words of encouragement)
Reputation	the perception that project members hold about another member's character, knowledge and skills	Social expectations: Importance of fulfilling social expectations
		Individual expectations: Importance of fulfilling individual expectations

Application of Restricted Access: *reducing number of key members selected for certain types of exchanges across project team*

The application of restricted access in the Lake Project network is identified through constructs including embeddedness, relational (social) contracting, status maximisation and ideological similarities.

In conclusion, I found that the amount of restricted access that exists in the Lake Project through strong embeddedness among the teams, relational contracting and tendency towards status maximisation in all locations is enough. This means the existing restricted access in the Lake Project team is not too weak to complicate coordination of complex and difficult tasks and it is also not too strong to reduce the teams' motivation for innovation and quality tasks which might lead to low performance in the project team. The existing restricted access made the team members interact with fewer team members for certain types of exchange more frequently. More focussed interaction with fewer team members not only makes it easier to communicate and interact but it also reduces the risk of opportunistic behaviours occurring.

I found that interactions among the Lake Project teams are restricted for certain types of exchange. For example, the Lake Project has limited developers and customer interactions; if developers need to interact with the customer, they will get to product owner and e-commerce representatives. Indeed, as part of agile environment setup of the project, business representatives (product owner/e-commerce) are nominated as a point of contact for duration of the project and the team members only interact with these representatives and not any other potential user. *"…IT people never interact with the customer, the product owner and e-commerce people are real representatives from the customer for us."* (The Lake Project team leader)

Another example is that testers and developers do not interact with e-commerce if there is an issue with requirements. Instead they go through BAs across locations as they are responsible for this type of exchange between e-commerce and the core team members. On the other hand the Lake Project BAs have access to e-commerce representatives who are only allocated for this project.

I found there is a high level of embeddedness among the Lake Project teams locally and across geographic boundaries. This embeddedness is stronger among team S and team C and even stronger among developers across locations. I also found the existence of strong ties among the teams facilitates stronger levels of commitment and trust in regards to relational contracting. The Lake Project teams across locations involved in relational contracts try to work with each other to ensure long term benefits and to fulfil the other teams' needs and requirements. This also facilitated safeguarding and protected the project teams' exchange and interactions across locations.

In addition, I found the status maximisation strategy restricts access in the Lake Project because the project team members are interested in interacting with similar status team members more frequently both locally and across geographic boundaries. The status maximisation strategy is facilitated by structuring the project into different levels and by having similar status forums and meetings such as Showcase meetings and Homemade Jam meetings.

Furthermore, I found that it is easier for the Lake Project team members to communicate and interact with the other team members who have similar interests. This is facilitated by having forums for the project team members across locations with similar interests such as Developers' Forums and Business Analysts' Forums. The Lake Project team members found it easier to have more frequent interactions with the other team members who have a similar nature of work. For example, I found that team S and team C have a tendency to have more frequent interactions with each other as front end developers are closer to team M as mainframe developers. I also found team C has a tendency to communicate more frequently with the other team members who speak the Chinese language. Finally, I found that regular interaction among the project team members who have ideological similarities enhances the coordination among them. The following sections set these findings out in more detail.

Embeddedness (Jones et al. 1997): *The extent to which strong ties among project team members exist*

I found strong embeddedness exists among the Lake Project teams locally and across geographic boundaries. One of the reasons for strong embeddedness is the existence of a variety of interactions in the Lake Project such as forums. Forums for each group helped to familiarise themselves with the other team members' skills, capabilities and personalities. The high level of embeddedness facilitated fewer, more focussed and frequent interactions with other project teams locally as well as across geographic boundaries.

It is especially critical for distributed project teams like the Lake Project to have this image of the other teams because of the large number of team members involved in the project in addition to the distributed nature of the project. This knowledge and familiarity with the other teams makes restricted access possible because information about the other teams is transferred. This makes it much easier for the teams to know whom they need to interact with for certain types of exchanges. It also safeguards their interactions because the knowledge they have from the other teams allows them to get to know whom to avoid interacting with. The following paragraphs provide more detail.

There is a strong bond among the Lake Project teams although they are distributed in three locations. According to the Lake Project IM the tie among the teams across locations is relatively high. He believes the reason is that the Lake Project team is practicing agile values "*…I mean agile is about collaboration, high level of collaboration, a lot of interactions and communication. So those values drive that kind of behaviour,"* he said.

Although most of the Lake Project teams are distributed geographically, the teams are quite proactive at keeping the bond and relations with other teams across locations. For example, product owners and stakeholders are distributed as well as the core team members. Such distance usually make the interactions more challenging and extra effort is needed on behalf of all the teams to make sure that they maintain the relationship and the bond. *"…we are finding it at this stage where we were trying to discover the requirements, the bond and the communication that happens between everyone is quiet strong."* (BA, Team C)

The project manager believes that one of the reasons for the strong bond between the Lake Project teams was the fact that they had to face a lot of challenges together in order to make such a high profile project deliver. What made this project more challenging compared to other projects was that the Lake Project team were the first team to work distributed in an agile environment and the first team to work with a partner in China. *"I found in other projects as well, the more challenges you face, the stronger the team bonds. So we do have a support network between the team members in the Lake Project if things are getting too frustrating."* (Project manager, Lake Project)

The ties between the core team members of testers, developers, business analysts and the IM tend to be stronger than other stakeholders because they communicate with each other more frequently in terms of ensuring they are developing the right product. One of the interesting points about the ties between developers across locations is the fact that they are distributed causes the bonds to be tighter because there is a strong sense that they have to make sure they are constantly pairing while distributed. In fact the pair programming practice across locations became very dependent on the bonds because they needed to maintain the close communication and frequent interactions.

Another practice that enhanced the strong ties among the project teams was co-locating team S, team M and team C. At the beginning of the project, all three teams were co-located at Sydney for six weeks. This was a great opportunity for all the teams who had never met each other to build the initial ties. They were able to meet the business people and all other stakeholders, participate in workshops and become familiar with project context. Everyone had a strong sense of who the other teams are before they went back to their original location. This co-locating practice was repeated a couple of times during the project life time by co-locating the three teams in China and Sydney.

On the other hand, although there is a high level of bond among the teams locally, the level of bond among different teams across geographic boundaries is different. When comparing strong ties across the teams, team S and team M developed an average bond due to the nature of their work because team M is focussing more on mainframe platforms where team S and team C developed higher ties because they are both user interface focussed. Therefore, more communication and frequent interaction among the teams with similar nature of work caused higher level of embeddedness. The Lake Project team leader score the ties among the teams as *"…so from scale of 1 to 10, when I say strong ties in Sydney it would be 10 and the same in Melbourne and China, between Sydney and Melbourne it would be probably at the start I would put it about 3 or 4 and now is probably 6 or 7. For China it would have been 0 because they never met, now between Sydney and China is about 9. Melbourne started stream working with China in June so they are around 9 as well."*

An example of existing of strong tie between team S and team C can be found in the flow of work that goes between BAs, developers and testers. A piece of work could be analysed in China and then given to someone in Sydney to work on. Or it could be analysed in China and given to someone in Sydney and then tested back in China. Therefore, for a piece of work to be completed, the important part is about who is available and has the right skills to pick up a particular story card. In fact, the Lake Project has a cross functional team where everyone is relying on each other *"…so a mix of tester, BA, developer and business SME (Subject Matter Expert) who does what isn't dependent on the geography, it's just more about where the work goes. It's not a virtual team, it's a one team. The only challenge about the geography is about the communication and the technology we rely on to have the right conversations. The way the work goes isn't depending on where the person is located."* (Team leader) The strong bond and high level of embeddedness among team S and team C is obvious in this example.

Relational Contracting (Jones et al. 1997): *Project team members don't agree on a detailed plan but on common goals and objectives*

Based on my analysis, the relational contract restricts access by making it accessible to the Lake Project core team members only. It also facilitates safeguarding by setting the expected values, objectives and behaviours for the teams, as is discussed below.

The Lake Project core team members agreed to co-operate with each other to establish a long term relationship that is guided by a relational contract among them. The Lake Project team leader set out the relational contract with the whole core team when assembling the Lake Project team at the beginning of the project. This relational contract is designed and accessible exclusively by the core team members, meaning the relational contract restricts access in the Lake Project by making the team members' exchanges and interactions more often with fewer team members involved. The relational contract is about setting what values, objectives and behaviours are expected from the team members. It provides boundaries for the team members in all of the teams to interact and exchange information more often with those who are part of the contract. In this way, the relational contract protects the teams from possible opportunistic behaviours by the other project teams.

The iteration manager provided the Lake Project team relational contract as follows:

- Keep on time to meetings
- Code peer review before SYST – Protect
- Be mindful of people's time when booking meetings
- Ensure meetings have relevant outcomes and people

- Face to face when possible
- Keep good time
- Every review is valid
- No meeting at lunch time
- Freedom to withdraw if unnecessary
- If it needs a brain it needs a pair
- No check-ins on red build
- Do not leave broken build overnight
- Frequent tech huddles
- Course correction not course perfection
- Keep JIRA up to date

The strong ties among the project teams helped teams across locations to have a better understanding of each other and have regular interactions. Through these interactions, the teams regularly talk about the relational contract if they need to make any changes. For example, one of the points that the team members understood about the other team members across locations was that no meeting at lunch time was set for the Lake Project teams because of the lunch time restriction in China. Team M and team S learned that if team C members do not get their lunch at lunch time in China and get busy with meetings, after that there will be no lunch left for them to buy. Therefore all the project team members decided to include this term in relational contract so everyone become aware of this and respected it. Since then there was no meeting at lunch time (China time) and team C members did not miss their lunch.

The other example concerns the broken builds. Because the Lake Project worked in incremental build settings and considering the time zone differences between the Australian teams and the China team, it was critical to fix any broken builds and codes for the day before leaving the office. Team C was two hours behind team S and team M which means they do not come online until around 11am Australia time. So, if a broken build was left overnight and team S were picking up the broken build at 9am, they were not able to have a conversation with team C about the broken build until team C had started work some two hours later. It would make the coordination much more challenging. This made the project team include a ‘do not leave broken build overnight’ term in the relational contract so that all the teams across locations were required to follow it. Indeed, this is a safeguard strategy to protect the teams' interactions in all locations.

Finally, I found that regular interaction among project teams who are ideologically similar enhances the coordination among them. The above examples indicate how the Lake Project team members in the geographically separate teams are committed to co-operate with each other and meet each other's requirements. Despite the relational contract not being legally binding, the team members did not take advantage of the non-legality of the contract and did not behave opportunistically. Instead, they felt they had a social commitment to the other team members through this contract. The Lake Project IM explains that *"sometimes teams implement the fines as part of the relational contract so if you break the social contract you have to buy cake for the team or something like that. When we put the social contract together I don't think anyone broke it since."* This indicates that the relational contract somehow safeguards and protects the teams' interactions across locations. The Project Lake teams try to keep the relational contract short and clear so it is understandable for all the teams across geographic boundaries.

Status Maximisation (Jones et al. 1997): *Exchange and interact among members of similar status and avoid frequent interaction with lower status*

Analysis of the data indicates that the Lake Project teams have interactions and exchanges with other similar status team members on a regular basis. This restricts access because the project team members have frequent interactions with fewer team members. The Lake Project team members found it easier to interact with fellow project team members of similar status. To ease the coordination and reduce the communication overload, the project is structured with different responsibility levels such as program level, project level and the core team level. The project team members at the same responsibility level find it much easier to interact and exchange with each other. For example, it is much easier for a portfolio manager to interact with program IMs and project managers instead of interacting frequently with the team leader. The Lake Project team leader reports *"the portfolio manager does not need the team leader to deliver the project for the portfolio manager. He's already got project managers and program IMs that would deliver the project. He needs very little contact and as a matter of fact if they get every possible contact from everyone, they got overloaded."*

At a social level, senior managers in the core team level prefer to interact with the project team members if any issues arise. However, in terms of work related interactions, they prefer to go through the structural channels and interact with the same level managers. Another example of status maximisation is visible in the Lake Project regular meetings such as Stand ups, IPM (Iteration Planning Meeting), Showcase meetings and Homemade Jam meetings. Each of these meetings are restricted to team members of certain status in the project team. For example daily Stand ups and IPM (Iteration Planning Meeting) are restricted to the core team members and IMs (core team level), with the team leader and the other project team members from the project and program level being excluded. Similarly, Showcase and Homemade Jam meetings are restricted to different project team member levels (Section 3.6.3). The main reason to hold Homemade Jam meetings is to have similar status project team members (core team) separate from other team members' status (senior managers) in Showcase meetings. In fact, the Homemade Jam meetings are really another version of a Showcase meeting that is created exclusively for the core team.

The way Showcase meetings were originally structured and presented was co-opted by management at the program level because the Lake Project was part of the program A (Section 4.2.2). In Showcase meetings, there were a lot of pressure and influence from senior management (program/project level) about how they would like Showcase meetings to run, what they need and how to present. For example when a core team member was presenting a particular area the senior manager found it disruptive and said there is no need for them to go through these details, individual story numbers and implementation phases. They found it waste of the time for people who are attending.

Another reason that senior management found the core team participation disruptive was due to the geographically distributed project teams. The Lake Project IM reports that "*the instruction that we've had from the senior stakeholders is that it is disruptive when we have people speaking from multiple locations. We would rather have one person who knows the material to present to the staff."* In addition, team M and team C had to be connected through technological communication tools which were, at times, unreliable and sometimes failed during meetings. The Lake Project senior managers had limited time to dedicate to the Showcase meetings and wanted to maximise the benefits of the meeting and minimise the interruption.

On the other hand, there has been a strong feeling in the core team level that the Showcases are not effective for them. The core team members wanted to have Showcase meetings for the team with details to collect honest feedback in a 'safe-to-fail' environment about what they have achieved in a given iteration, what they have not achieved and what their risks and issues were. That was more a status update that to the program and project level management. The core team level project team members believed that Showcase meetings were owned by program/project level team members. *"So for all purposes, that Showcase for senior managers is not a Showcase in the agile sense of the word, in the way that the team would own the showcase. It is a status update that the IM needs to deliver to the senior management."* (Developer, team S) Therefore a decision was made that, Showcase meetings were not an effective use of the core team's time and the team would send a representative to that meeting to present the way senior management would want.

The core team members wanted to have Showcase meetings that they could own without having the senior management level of influence where they can present what they like in a way that benefits the core team (technical people). That is why they came with the idea of having Homemade Jam meetings. Homemade Jam meetings are similar to the Showcase meetings but this is exclusively for the technical team members of any project organisation-wide to meet up and to interact with the other technical team members that have similar interests and status. Representatives from all the different Ocean Group IT core teams across locations are invited to participate in Homemade Jam meetings. "*What's very different about it is that the core team has total control of what's going to be presented, the owners of the Homemade Jam are team members where the owners of Showcases are the senior stakeholders and when they do say, don't present this way, don't present this information, we have to. We are not the owners of that ceremony."* (BA, team S)

In the Homemade Jam meetings, the team explains the goals they are trying to achieve in the Lake Project. They have been trying to listen to feedback from other participants such as feedback about how they can achieve their goals and at the same time if there is any feedback about what participants would like to see or not see in Homemade Jam regular meetings. In this way, restricting the Showcase meetings to different project team members' status (program/project level team members and core team level) made it much easier for the project team members with similar status to interact and exchange across geographic boundaries.

Ideological Similarities: *interactions among members with acceptably consistent and similar interests*

The tendency towards interaction between the team members with similar interests is visible among the Lake Project team members across geographic boundaries. For example BAs have less interaction with developers not because they do not have access to them but because from a BA perspective it is easier to go to BAs, explain and give context. Team S BA reports that *"I prefer to have more frequent interaction with BAs across locations because it's easier and it's probably more necessary. The BAs spend a lot of time communicating across geographic boundaries and the vast majority of the questions can be answered by BAs."*

The Lake Project team members prefer to interact more frequently with the other team members who have similar interests because this is seen to avoid confusion, save time and be more focussed on their exchange. This goes some way to explain the existence of a range of online forums such as the BAs' forum; the Developers' Forum, the Testers' Forum and the Technical Forum. In these forums, the team members across locations get together, communicate, share information, transfer knowledge and discuss current issues. It's a daily Stand up for 15 to 20 minutes throughout the day which is only for the team members from a specific discipline and interest across geographical locations through Skype.

I also found that the team members with similar natured work tended to interact and exchange more frequently than those team members with different natured work. For example, team S and team C developers were both front end developers where team M developers were more mainframe developers and it was evident that team S and team C developers preferred to interact more with each other than with team M developers, as explained by a team S developer: *"Sydney and China are front end developers and because the nature of their work is the same they have high interactions every single day. But because the nature of the work in Melbourne is more mainframe, the daily interaction is not as high as Sydney – China."* (Developer, team S) In this example, the team members of both teams found it easier to communicate due to their similar working nature and it enhanced the coordination among the teams across locations.

Finally language was another driver for team C members to tend to communicate with the other team members who spoke Chinese as well. Although team C members participated in English language courses as part of their job requirements, often language barriers made it challenging to explain complicated issues or transfer the context to team C members. For example, the context of the 'Green Slip' or 'Pink Slip' was difficult to communicate fully in English to team C members and they had it explained fully in Chinese through their BA. *"…especially in China when there is complex subject that needs to be communicated, team members preferred to go to their geographical representatives [such as BA] and asking them to explain, give context and let them to re-communicate it in Chinese for them."* (BA, team C)

The above examples indicate how the Lake Project team members are more likely to have frequent interactions with the other team members who have similar interests. The tendency to communicate with the other team members that have ideological similarities restricts access in the Lake Project as the team members across different locations choose to interact with fewer team members regularly. I also found that ideological similarities improved coordination among the teams much better across geographic boundaries.

Application of Macroculture: *common norms, values, goals and assumptions that are shared across project teams*

I found that a strong macroculture exists in the Lake Project team both locally and across locations. The macroculture provides a similar way of working on all locations which enhances the coordination of exchange among all the teams. In addition, I found some of the shared norms and values act as safeguard and protect the Lake Project teams from possible opportunistic behaviours locally and across locations. The following paragraphs provide more detail.

Macroculture is concerned with shared norms, values, goals and assumptions comprising technical or professional knowledge that guide actions and create behavioural patterns among partners. The application of macroculture in the Lake Project network is identified through constructs including shared common norms, shared common values, a shared common set of goals and shared common assumptions (See Appendix C).

At the beginning of the Lake Project, like any other projects in the Ocean Group, the team members were sourced from different domains and at the end of the project, each team member will go back to that domain. Sourcing the team members from different domains benefitted the project because the team members brought different backgrounds, ideas, suggestions and different ways of doing things with themselves to the project. On the other hand, as explained before, there is a high level of embeddedness existing among the project teams across locations. I found that such structural embeddedness in the Ocean Group plus the strong ties among all the project teams across geographic boundaries allows the inter-company movement of the projects' team members.

In fact, the structural embeddedness facilitates diffusion of norms, values and expectations among the teams in all locations (Jones *et al.* 1997). This enables business representatives and the project team members to share a common understanding of the norms and values and strive to perform at the highest level. The shared understanding enables the delivery of software to the satisfaction of the business *"…they usually deliver within the scope they promised to the customer*" (Lake Project manager). The norms and values provide the business representatives and the team members with behavioural guidelines for their interactions. It also facilitates Lake Project teams sharing a similar way of working on different sites.

I also found that shared norms, values and goals are encouraged as part of each of the project team members' KRAs (Key Responsibility Areas). Each employee in the Ocean Group has certain targets they need to achieve by the end of the year. Those target points are actually based around the River values and have been implemented to ensure that what needs to be done is done and are designed to encourage the team members to achieve the targets of their role. For example, as a developer the KRA might be to develop quality code and you need to demonstrate that. BA from team S adds *"It's important to have KRAs because they are the primary responsibilities of an individual, the core area which each person is accountable. So it's important to have them because they set goals and objectives for each team member, prioritise their activities to improve their work management. It also helps to clear each member's role so their role's purposes can be communicated to others."* KRAs are there for the team members so they can make value-added decisions and focus on their achievements rather than activities.

I found that shared norms and values between the teams across locations were not equally strong throughout the project. For example, the Lake Project IM explains that shared norms and values in the beginning of the project between Sydney and China teams were much more closely aligned than they were with Melbourne team. They also noticed that, in terms of the value that the team members put on pairings, Retrospective or immediate feedback or collaborative working, this was much closer between Sydney and China teams rather than between Sydney and Melbourne teams. This appeared to be the case regardless of whether that was within the co-located team or across the locations. It would appear that this is due to the way the Melbourne team operated as the Melbourne team was focussed on mainframe development, so there were considerable differences in the technical work when compared to team S and team C. However, shared norms and values strengthened between the Melbourne team members and the other two locations as the project progressed and moved on. By way of example, a team S developer explained that *"…they [team M] much preferred to work in three week iterations or longer if they can get them and they tended to try to play stories as large single pieces whether it might takes three weeks or four weeks to do".*

There were also lots of efforts to harmonise team C members' culture with Australian working culture as general and Ocean Group norms, values and goals in particular. They have gone through a lot of lessons, classes and extracurricular activities on the site apart from their usual work. In order to make team C more harmonised with the Ocean Group teams (team S and team M) and as part of the required quality to be a member of team C, River chose people who were comfortable expressing their opinion. These particular members were considered very open team members for this project. The Lake Project agile coach explained that *"one of the classes that they go to are English classes but they also learning about Australian culture and knowing about how people work here and what the working environment is like. Plus they also pick it up while they are working with us, interacting and communicating with us."*

Through the analysis, I found that a strong macroculture exists in the Lake Project team. Shared common norms exist among the teams locally as well as across locations. I found that some of these shared norms are part of agile practices, such as pair programming, daily Stand ups, Retrospectives, IPM (Iteration Planning Meeting) and code review; and some are not agile practices, such as shoulder checking (peer review), shadower role, body role and forums. I also found shared norms like shoulder check, shadower role and buddy role are used as safeguard strategies for the Lake Project teams to protect their interactions and exchanges both locally and across geographic boundaries. For example, having constant shoulder check between developers, BAs and testers lessens the likelihood of any opportunistic behaviour.

The shadower role and buddy role give the flexibility to the project teams to have an internal team member to replace the role instead of introducing a completely new person to the project.

When a team member cannot be present (is on leave, ill, or leaves the project) there is another team member to take up the role. Therefore, the project teams are protected and the likelihood of opportunistic behaviour is reduced. Shadower and buddy role norms also make coordination of new project team members with the existing team members (across all locations) much easier and faster because as the shadower or buddy, being an existing team member, is familiar with procedures, values, norms, goals and context of the project.

Some other norms have been set by the project teams themselves as part of their working standards such as story writing standards and use of JIRA software. I found that using JIRA as an electronic agile wall provided transparency of the teams' activities in all locations and this visibility reduced and safeguarded against opportunistic behaviours throughout the project. It also makes it much easier for all the project teams to coordinate across locations because of the visibility and clarity of the teams' activities that the software affords. There are also norms that are developed through frequent interactions among the project teams locally and across locations, including norms relating to using a common language, approaches, terminologies and strategies.

The common strategies, such as avoiding the 'blame game' and voicing out issues, shared among the Lake Project teams locally and across locations reduces the possibility of opportunistic behaviour. In fact, these strategies safeguard the project team' interactions and exchanges locally as well as across geographic boundaries. When all project members have a shared understanding that, for any problem to be raised, there will be no blame gaming, a trusting and safe environment is provided for them to voice out issues as soon as possible. Therefore, these strategies not only protect the project teams from unacceptable behaviours through the project life cycle, but they also facilitate coordination among the teams. When issues are raised early without fear of being blamed, it makes taking care of the issues much easier and faster before they cause complications in coordination among the teams. I also found that shared norms are shared at different levels in the project. For example, norms like pair programming are shared at the team level as well as at the program level.

The Lake Project teams share common values which are a mix of the Ocean Group values and the River values. These values are very similar, such as honesty, courage, fairness, respect, caring and trust. Some of the other shared values are agile values, such as the notion that individuals and interactions are more valued than processes and tools, the notion that customer collaboration is more valued than contract negotiation and the common sense notions of equity and fairness. Finally, other common values shared among the project teams in all locations are sourced from their social contract values. In addition, the Lake Project teams shared values through their socialisation activities. They were able to create a sense of mutual interest. One of the main values that the Lake Project core team members claim to share as a team value across locations is delivery of quality code. They work hard to keep this team value by delivering the quality code through safeguards they have in place such as constant shoulder check and code review. The existence of such shared values among the Lake Project teams locally and across geographic boundaries facilitated a similar working environment for all the teams, which enhanced and eased the coordination required among the teams in all locations.

In addition, I found trust as one of the most important values shared among the teams in all locations, with a high level of trust evident among the Lake Project teams. The high level of trust is not surprising given that it is a strong shared organisational value for both the Ocean Group and River, and is hence very much embedded in their work culture and daily activities. Developer from team S says *"One of our values is trust and we have been living that value very much. There is a lot of trust among the teams otherwise without that the team would not be able to work well."* This facilitated a trusting and safe environment for all the project teams across geographic boundaries that reduced the possibility of opportunistic behaviour among the team. In fact, a shared trusting value protects the Lake Project teams' exchanges and interactions through the project life cycle.

There are four major goals for the Lake Project and these are shared among the Lake Project teams in all locations at all project levels. These goals are: 1) delivering the project on time, 2) making the distributed agile component work, 3) continuously deploying into production and 4) delivering the software product to the satisfaction of the business. In addition to these four goals, the project also sought to achieve success sliders (refer to shared common set of goals, page 135). I found that, as part of the agile context in the project, goals that are set at the project level are driven by project success sliders and it is an indication that all project teams are aiming to achieve this project.

The Lake Project teams share certain assumptions across all locations. For example, the team members assume and trust that BAs, IM and e-commerce representatives communicate the same messages to all developers and testers across geographic boundaries. The Lake Project IM explains that *"…So there is a lot of trust amongst the teams to deliver the task, to assign the task to them, to complete the task on time, etc. For example, when we assign a task to a team member, we trust that the person completes the task on time. Yes, it's all about trust, even in China although they were new addition to the project team, it is the same because trust is a shared value between the Ocean Group and the River."* There is also a common understanding and assumption between team S, team M and team C that they are all expected to either participate in all project team gatherings and activities or challenge them, but as a Lake Project team member they do not have the flexibility of not participating. The Lake Project team members tend to work at a maintainable rate to keep a work life balance. *"…we always encourage and support the team members to maintain work and life balance"* (Project manager, team S).

I found that there is a shared assumption among the team members across geographic boundaries that unless someone is blocking the other team members to continue their task, they do not work until late hours. They all share the assumption that there is no weekend and late hours work required. I also found that working in an agile way is a strong assumption shared among all the project teams across locations. The assumption already existed in the River and the Ocean Group as their way of working and after the Lake Project teams assembled, they brought this assumption with them to the project. Indeed, agile is the way the Lake Project teams work because there is no other way for them to work. An interesting point that I found was that, due to different levels of awareness of project context, levels of assumptions that the project teams made across locations were different. Because team S and team M existed of Ocean Group employees they were more aware of project context when compared to team C members. China team members' assumptions were very different but this was aligned through the co-locating of team practice at the beginning and through the project life cycle.

In total, I found that there are many shared understandings and assumptions of the way the project teams work across all locations, which facilitated their interactions locally and across geographic boundaries much easier. Thus the shared assumptions enhanced coordination among the project teams. The following sections set these findings out in more details.

Shared common norms: *set of repeated behaviour expectations that individuals perceive from repeated attitudes, patterns or behaviours of others*

Shared norms exist among the teams across locations. Some of these norms are part of agile development methodology practices that are shared between all the project teams across geographic boundaries. Some of the norms are developed through frequent interactions among the teams like using common language, approaches and terminologies and some other norms have been set by the project teams themselves as parts of their working standards. In addition, holding regular daily Stand ups, Retrospectives, Showcase meetings (part of agile tools), IPM (Iteration Planning Meeting) and different forums (part of the project settings) became norms among the Lake Project teams.

On the other hand, some shared norms are shared at the team level, for example the Lake Project team always do pair programming and pair development. Some of the other norms are shared at program level, for example, the norm is that the Lake Project team have to work with the other project teams within the same program to get them across and synchronise the activities with them. The Lake Project team leader explains that *"The way we do our analysis, the way we do our development, we need to communicate them all with other project teams within the same program and make sure that whatever we do is standard of the project and we don't need to change them."* It is norms such as these that the teams built and then maintain throughout the project.

Shared common agile practices

The Lake Project manager believes that "*there is definitely a culture in the team across locations that everyone commonly agrees on certain norms like agile practices."* He explains that the existence of the culture of agility in the project sets the tone for the way the project teams work on different sites and share norms. For example, the Lake Project teams started sharing different aspects of agility and practices like pair programming across locations via MOC (Microsoft Office Communicator - an instant messaging tool), code review between team C and team S and some other processes shared across all locations such as Retrospective meetings on regular basis, Showcases and daily Stand ups through Skype.

As the Lake Project teams follow the agile methodology, everyone has an expectation across locations that they should have a Retrospective meeting at the end of each iteration. *"What you find is once those meetings get pushed out there is someone who shouts out and raises the questions that are we going to have the Retro soon? It's the same case for Showcases as well."* (Developer, team S)

Daily Stand ups and pair programming are other shared agile practices across all three locations. Pair programming is a norm that, despite the additional difficulties of doing it (i.e. the pairs could be in different locations), has a value to the Lake Project teams and has to be done. The pair programming made it possible for the project teams to complete the tasks that they were not able to do so otherwise. For example the BA from team S explains that *"We do have that front end developer in Melbourne and he is very reliant on pair programming because there is no one else to help him out, especially when it comes to context related business logic rules he's got to rely heavily on the distributed team."*

Code review is another agile tool that has become a norm between team S and team C. There is a norm among developers in Sydney and China that at the end of each day or at the beginning of the next, they informally sit and go through each other's code and review the codes that were checked on the previous day. The Lake Project IM adds that *"there are a lot of other things shared between Sydney and Melbourne like the way we write stories, the language that is used, how meetings are run like Stand ups, how Retrospective is run is always the same; these are part of the agile tool sets, like a planned one."* Such shared norms facilitate similar way of working across locations. Similar ways of working across locations facilitates coordination among the project teams on different sites.

Shared non-agile practices

Measuring tasks

One of the non-agile practices that are shared between the teams as a norm is measuring the number of days people work on a task. The norm is to measure how close the team members are to completing tasks and this is shared and done across locations. This is not necessarily an agile practice, it's something that goes on throughout the project. *"So we have a common approach, like if something is in 25% you can possibly have another developer jump on it and* *pick up something new. It drives a lot different conversations and improves the coordination among team members on different sites."* (IM, Lake Project)

Shoulder check (peer review)

Shoulder check is a shared norm that the Lake Project team members do whenever they need to, in all locations. The communication with the other team members is pretty constant across the team, so the team members spend a lot of time doing shoulder checks with developers, testers and the business.

The norm is to have one shoulder check before developing the story and another one after development is done and before sending it for the test. So when there is a story and it's ready to be picked up by a developer, a team member does a shoulder check where a BA sits with the developer and ensures that they understand what that story involves and what the expected outcome is when the development is done. When the development is done and the story is ready to be handed over to the tester, there should be another shoulder check between the developer and the tester and the BA. Developers would talk to testers and explain what they've done and what checks need to be done. Shoulder checks are shared and are exactly the same between all the three locations. "*Shoulder check works well where developers and testers are all in Sydney or China* [co-located] *and also worked very well across locations."* (BA, team C)

Based on the above discussion, it is clear that shoulder check is used as a safeguard strategy that is shared and used by the project teams locally and across locations to protect the project teams from possible opportunistic behaviour.

Shadower role

A norm that is common and shared across locations among the project teams in general, and developers in particular, is to have a shadower role. A shadower is a project team member who follows another project team member around so that they can learn what the project team member does. It is a form of learning by watching. After one week the shadower and shadowee switch roles with the shadower doing the work and the shadowee watching. A senior developer in Sydney explains: *"All project roles have a shadower, projects tend to have one or two* *shadowers around on frequent basis I mean within any given time so far we've had at least one or two shadowers."* (Developer, team S)

Shadowers are additional resource for the project and are a backup role for the main role they shadow. This is considered very important for roles such as the core developers. They are essentially a partial resource because they are there to learn from the others, but also contribute wherever possible. This norm is used as a safeguard and gives the team better flexibility if a person is on leave or gets sick - there is always an extra person to come in and fill that role straight away because they have been there watching how things are done.

This norm is important as it assists in minimising the impact of team members leaving during the project because it allows for replacements for the roles to be brought up to speed in the project much faster with rest of the team across locations.

Buddy role

Another shared common norm across the project team is buddy. This norm is mostly used when there is a new team member to the team and they want to fit the team member in to the project team. The new team member starts working on a task while they have a buddy. The buddy is usually a senior team member that sits next to new team member. They assist the new staff member to integrate quickly into the team and get an understanding of what's happening so they can start bringing value and bring new ideas to the table. Buddy is there to provide support to new team member; he guides and checks on him on daily basis.

Forums

Forums are set norms for the Lake Project that are shared between team members across geographic boundaries. These forums are like catch up sessions for the different groups of team members. Examples include the Testers' Forum for testers, Tech leads' Forum for tech leads, BAs' Forum for BAs and Developers' Forum for developers in Sydney, Melbourne and China. The Lake Project teams have these forums on regular basis. For example, a shared approach among developers across locations is to have a forum session whenever a big decision needs to be made on technologies or major decisions about which path to choose in terms of how to build the applications. BA from team C says *"it's become a norm to jump into a forum session and have a conference to just discuss how they are going to resolve the issue."*

Shared common working standards

There are certain working standards for developers, testers and BAs that are shared across all locations and agreed to at the beginning of the project. Any required changes to these standards during the project will be discussed with the related group, such as testers or BAs. These standard procedures are used by the Lake Project teams as a norm while they are doing their work, such as BAs' story writing. *"…so we have agreed in phase one that we will write the stories in certain way. If we want to change this norm of story writing then we discuss it between ourselves (BAs) and say 'hey let's do it this way' and we pass it on to few developers and testers across locations to see whether they are OK with it"* (BA, team). Similarly the developers have standards and norms that they must adhere to, and these are shared among developers in all locations.

Use of a shared agile electronic wall (JIRA) that needs to be kept up to date is another work standard that is a norm for all core team members. JIRA is a tool that is used to provide visual representation of tasks and their status. It is used during daily Stand ups through screen sharing sessions across locations to talk about the works in progress. All the teams have visibility through these processes and share this practice. The transparency and visibility of activities that JIRA provides for the project teams across all locations not only protects the Lake Project teams' interactions and exchanges but it also enhances the coordination among teams too.

The teams also share a sense of mutual interest across geographic boundaries. For example, any sort of negative attitudes or comments have been stopped in the Lake Project and are not encouraged by the rest of the team members. Negative gossips, comments or attitudes are rejected by the other team members and brushed off. The Lake Project team leader adds, *"We acknowledge positive behaviours during Retrospective and thank team members for positive attitude. This becomes a norm in our regular exchanges across locations that negative attitudes or comments are stopped by team members."* Encouraging such shared norms across the teams safeguard the teams from any inappropriate and opportunistic behaviour.

Shared common strategies: Avoiding the 'blame game'

There is a shared norm and expectation between the project teams that anytime an issue has been raised, it is so it can be improved and made better for the entire team. It is not so that fault can be assigned to a particular person or so that a specific team member can say that I'm raising this but it's not my fault. In fact without having that kind of shared culture across all locations, there would be a lot more conflict among team members.

The Lake Project IM explains that "*it does sometimes exist where this shared understanding is weak and we have had conflict in this team, things get raised either in confrontational fashion or in a kind of blaming fashion."* However, where blame avoiding fashion is stronger, there is a more trusted environment so the teams have a clear understanding what could have happened. Therefore, it lessens possibility of opportunistic behaviours and the interactions among the project teams are protected.

Shared common strategies: Voice out issues

One of the common strategies that was encouraged among all locations was to voice out any issues as soon as possible and not to hold it back until it's was too late because *"bad news doesn't get better with age"* (team leader, Lake Project). The project teams are aware that they need to make any issue or risk known early and quickly to their managers and communicate it to as many stakeholders and project managers as possible. Retrospectives are definitely one of those times where everyone in the team across locations can have their say as to how they think the project is going and what they can do to improve. In addition, it encourages the team members to feel safe to voice their opinion. *"This becomes like a repeated pattern* [norm] *for team members during team socialisation like Retrospectives."* (BA, team S)

The Lake Project team leader kept repeating this point so that it became a norm of the project members across geographic boundaries. The team leader repeatedly reminded team members that *"…if you have an issue don't wait for Retrospective to raise the issue, raise it in Stand up or when you've got everyone and then we can schedule something in or we can stay behind or after the Stand up and we can discuss it but raise it as we go through."*

It is for this reason that on many occasions the project team members felt free to raise issues as the project progressed. For example, a developer explains that in Stand ups, out of 15 iterations, there were seven iterations where team members felt empowered to speak up about the time things were taking. It's interesting that *"…a lot of time people very openly express their ideas and say 'I really don't think we can finish this let's have times up or times down' or 'we're really looking tight here' if you feel we should do it."* (Developer, team S)

Another example mentioned by the Lake Project IM was when they identified some issues about their integration with an external website. They raised them to project manager and said they cannot integrate with the site unless he provides them with definite interface. *"We informed the management nice and early here. It helps not only the team but the stakeholders to help out more efficiently."* (IM, Lake Project)

The above discussion and examples clearly indicate that common shared norms in the form of strategies such as voicing out issues act as a safeguard and protection for the Lake Project teams from unacceptable behaviours. It also enhances coordination because any issues are shared as early as possible and will be discussed before it gets too late and complicates the coordination among the teams.

Shared common approaches

Throughout the project, the teams developed and shared certain common approaches or tacit rules to particular situations. For example, the Lake Project IM explains that *"when we review our stories and find that there are too many points and more work to deliver by the delivery date, everyone knows what happens. You go to planning and you get the business into the room and working out our approach to it."*

Another shared norm among the teams is that nothing gets checked in (no code gets into the builds) without 80% code coverage. This norm is enforced; there is a tool that checks the coverage, it runs and checks how many lines of code there are and if you check in and it does not cover the 80% code coverage, it crashes. This norm is enforced and shared between all the three teams across geographic boundaries.

Team S developer talks about another example where *"when something comes back from test …everyone knows the common approach for the tester is they need to go and speak to developers and explain, rather than just say this is not correct and doesn't work and it has a bug or raise the* *bug and through it back it to the line. It's a common responsibility."* These shared approaches makes coordination much easier among the teams across locations.

Shared common language

The Lake Project teams developed a common language through a lot of interactions throughout the project to summarise complex routines or information. There are words that they use in their language that are not generally spoken as a general word like the word 'sync' for synchronisation, but they use it a lot. These words are shared among all the teams in Sydney, Melbourne and China and it becomes a norm among the teams to use them. *"…because we know and share a context, [we] form a language around that context."* (IM, Lake Project)

Majority of common language and terminologies that are shared among the project teams across location have been developed at the beginning of the project. Terminologies like IPM (Iteration Planning Meeting) and agile methodology terms like blockers (something that is waiting to be done), builds, stories, features and release plan are quiet common. I found that developing common language and terminologies that are shared among all the project teams, locally and across locations, makes communication and interactions easier and faster for team members therefore they enhance coordination among the Lake Project teams.

Shared common values

There has been a lot of effort in the Lake Project to normalise the values that the Lake Project team members work to. They managed to normalise them between the teams in three locations and get a similar set of values. They are becoming normalised but only through a lot of effort. For example, the Lake Project manager reports that *"team C from the River does a lot of work in agile space, they don't put people on projects that don't have agile experience or are not willing or passionate about working in agile fashion. That normalisation would always be easier because they all coming from the same values."*

The Ocean Group partnered with the River Company in the Lake Project to source team C in China. The Ocean Group had worked with them for years on different projects in Australia but not overseas. However, the reason that the Ocean Group chose them was they were familiar with their way of working and were aware that their values are very similar to the Ocean Group values. The Lake Project manager explains *"…we kind of noticed that's how they work in Sydney and the U.S. and they reassured us this is the case in China also. We knew the way they worked in other countries even though we never worked with them in China."*

In addition, one of the main values that was shared between the Ocean Group and the River (and drove the Ocean Group to choose the River as partner for the China team) was that the River was focussed on delivering to the satisfaction of the business. The Ocean Group were aware that the customer was very important to the River and customer was a driver. That was one of the biggest reasons that the Ocean Group chose the River as a partner. At the same time it reinforces that the Ocean Group as a company are really driven by the customer as well. BA from team C explains *"I think it was part of the culture to make sure for both, the Ocean Group and their partner to make sure that at the end of the day they deliver something for the customer at the fastest possible way and the most efficient way. So it was really driven for the customer."*

In River, all employees are familiar with agile and they were taught agile values as part of their recruitment process. Similarly, the Ocean Group is agile value focussed and most of their values and principles come from IT division-wide principle and values. Team C would fall under the River values, where a lot of values are common and shared with the Ocean Group values.

In general, the Lake Project teams agree on common values (Appendix H) such as the Ocean Group and the River values, agile values and social contract values, which are very similar. However, one of the main values that the Lake Project core team members shared across locations is delivery of quality code. "*We share the Ocean Group values and social contract values like honesty and trust. Beside this we also share agile values."* (Project manager, Lake Project) The Ocean Group values are soft values such as honesty, courage, fairness, respect, caring and trust. Such shared values across locations ease coordination as IM from team S explains *"…so the Ocean Group values make our days and how we work and coordinate straight forward."*

Other values shared among the teams across geographic boundaries are agile values such as to value individuals and interactions over processes and tools or value customer collaboration over contract negotiation (Appendix H). *"I mean agile is around collaboration and communication over process. The thing is that we have got a very large team which makes it very hard. So you* *do have people who work on one thing or one feature where they would collaborate more because they need to but at the same time they still need to collaborate with the rest of the team because they are on the same code base."* (Developer, team S)

The project teams share agile values through the shared understanding they have from these values. There are always shared agile values around, like communication and collaboration. For example, business analyst from team S explains "…e*veryone needs to have an understanding why we are here for whatever meeting it is or the Stand ups, why I'm coding, why I'm pairing with this person, everyone needs to have that shared understanding."*

Empowered and equal team members are part of agile team values and the Lake Project team members believe everyone is equally accountable for the project success. *"So this value is around one thing - that we are all accountable - so it means we are all equal as everyone has the same amount of influence on the success of the project because they have a shared feeling that they are all empowered members*." (BA, team C)

Relational contract values (Appendix H) also outline more of soft values. For example respect other the team members, no lunch time meetings to respect team C (because if they do not have their lunch within a particular timeframe all food will be sold out).

The Lake Project teams also share values through their socialisation activities. They were able to create a sense of mutual interest between teams across locations. One of the practices that helped in socialisation was to co-locate teams from different locations in Sydney or China for a period of time. During the co-locating practice, the project teams from different locations had the opportunity to socialise and have face-to-face conversation. Throughout this process, team members' discussions were more cultural and experience-based rather than purely work discussions, which definitely helped them to get to know each other better. It provided an opportunity for the project team members to start understanding and also to ask for more information in the right direction rather than just not having the context and struggling. The Lake Project IM adds, *"…they could ask more questions and be guided in the right direction but if the context was not there they really did not have any vision it would have been like working in a black box."*

Such practices facilitate teams across locations to share mutual interests during the project. For example, developer from team S explains that *"We had like little components that required constant monitoring and someone to keep an eye on it and we had a team member from China and a team member from Sydney that had mutual interest in that area and they were already aware of each other's expertise and interest".* They were both interested to develop their knowledge in that area. The team C member picked it up and they started to lead it together very well.

Another important value that is shared among the Lake Project teams in all locations is trust. Trust makes the project team members reliable. Developer from team S says *"One of our values is trust and we have been living that value very much. There is a lot of trust among the teams otherwise without that the team would not be able to work well."* The Lake Project IM explains that *"There is a lot of trust amongst the teams to deliver the task, to assign the task to them, to complete the task on time, etc. For example, when we assign a task to a team member, we trust that the person completes the task on time. Yes, it's all about trust, even in China although they were a new addition to the project team, it is the same because trust is a shared value between the Ocean Group and the River."*

The Lake Project manager also believes that there is strong trust among the teams across geographic boundaries. It is actually part of the working culture both in the Ocean Group and the River. *"I definitely believe that there is a high level of trust and strong ties between team members across locations and they are happy to work with each other. The real strategy for that is trust is part of the Ocean Group and the River values and agile values, so it's the way it works. I don't particularly think that we called it out in this project. It's probably because the Ocean Group guys are probably so used to working in this way that we probably assume knowledge and trust, the same case for the China team."* A developer believes that *"trust is all about knowing, well one aspect of it, is to know for example when I give you my story card I know that you tested it accurately or elaborated properly."* (Developer, team S) Predicted trust was observed as predicting what that person is going to do. So, when it is a shared value, then the positive attitude would be predictable and the teams would be able to build more trust.

As team C is part of the River, they have a similar value as well where they encourage honesty and openness and trust. So the harmony of having a safe and trusting environment across locations could be because of that. Also at the early stage and very beginning of the project, once the project team with China, Melbourne and Sydney was formed, they found trust and openness was already there. *"There was no need to encourage trust because it was already there. It was about establishing a relationship first, it was a positive and safe environment to start with."* (Lake Project, team leader)

The strategies that facilitate a trusting environment for the project teams to work with each other are embedded in the company's value list and include values such as to be honest and trustworthy. The project team leader and management at program level were frequently reminded and communicated these values to the teams. These values have become part of the teams' culture. *"It helps everyone else as well by allowing you to be open and talking about any risks and issues and raising them early. Yes, it's embedded in our culture where it's 'safe-to-fail'. This is the company level of values and it's the same and shared on the project team level."* (BA, team S)

The Lake Project teams went through different levels of trust during the project team's life cycle. These stages of the project are referred to as forming, norming, storming and performing. In the first stage you are forming the team and the project teams are getting to know each other. At this stage they are still not sure how much they can trust each other and how much they can be open with each other. The second stage of the project is norming, where the project teams are becoming more comfortable with each other and they start to socialise with and trust each other. This is an early stage of trusting. The third part of the project life cycle is storming where the teams start working with each other very well. They are storming and having fun, racing ahead and putting out code like there is no tomorrow. So storming is like the honeymoon of the project and then they get to performing where the teams start settling and this becomes a routine. *"In the Lake Project, team members' trust relationship was at the second stage at the beginning of the project, they were one step ahead of usual trusting phase. I would say it's because trust was a shared values in the Ocean Group and the River and it was part of team members working culture in all three locations. That is why they built trusting relationships with each other stronger and faster."* (Agile coach, Lake Project)

Furthermore, the existence of a trusting basis between the project teams in the very early stages sped up the project life cycle process and eased the communication and socialisation of the teams across locations. This facilitated the teams to trust each other more, to work better and be more open to approach each other. *"One of the things that I was passionate about at the beginning of this project was that there was a trusting base relationship between project team members and a lot of members shared my passion and support and helped me to build the team at forming stage."* (Lake Project, IM) This sped up the forming stage and its transition to the norming stage.

One of the practices the Lake Project team exercised to build trust at very early stages of the project was spending a lot of time getting project teams from all three locations into the same room. Most of the time they arranged to do fun activities that help people socialise, get into each other's personal space, 'break the ice' and boundaries between teams. *"I think one of the things that really helped is socialising, and socialising so team members become freer to speak out."* (Team leader, Lake Project)

Shared common set of goals

As part of agile context in the Lake Project, goals that are set at the project level are driven by project success sliders. These goals indicated what all the project members were aiming to achieve in this project. Indeed, one of the goals and aims of the Lake Project is to achieve as close as possible to the success sliders.

Project success sliders are another agile concept that the Lake Project team use at Ocean Group. The concept behind them is that these success sliders are the way for key project stakeholders or a product owner to convey their expectations to the project team in terms of time, cost, quality and scope. Each of these features reflects some dimension by which agile project success can be determined. As a general rule of thumb, all slider features cannot be set to 'on' because you cannot expect to be on time and on budget and have the best quality, even though that's the ultimate aim of the project and the project team will always try to achieve it.

Sliders show the project teams the stakeholder's (the business) expectations. They enable the team to come to a shared understanding around the expectations that lead to the success of the project. These expectations set the project goals that the project teams need to achieve. For example, *"If the business indicates that they don't care about the cost as much let's just deliver this just on time with a reasonable amount of quality for this product, in that case our goals for* *the team would be we can spend a bit of more money so let's gets more sources in so we can deliver on time and the quality."* (BA, team S)

Having the product owner and key stakeholders participate is critical to ensure team alignment to the success criteria as defined from the customer perspective. That is why the team should ensure that sliders are evenly distributed. *"We make sure that these expectations and goals are communicated to all three teams across geographic boundaries and they are aligned and not working cross-purpose."* (Team leader, Lake Project) The features used as success slider for the Lake Project are scope, cost, time and quality, alignment to technical strategy and delivery to business satisfaction. Therefore, the project goals are set around these features.

There are four major goals that are part of the Lake Project and are shared among all the project team s in all levels; program level, project level and the core team level. The Lake Project manager says *"We definitely share the project goals in all locations. The Lake Project team members are very well aware of the project goals and well working on the same direction towards achieving these goals in all locations."*

All the teams have the goal of delivering the project on time. This goal was very well communicated across locations and every single project team member put in an effort to achieve this goal. Business analyst, team C believes "*There is a culture of everyone commonly agrees on the project goals. Things like, we had a date that we had to hit for delivery and that was the goal. We knew what we had to finish and complete by what date and could not sleep."* A developer adds *"I think we were very much focussed and dedicated on the fact that we want it to be, every single one of us wanted every feature to be complete by the date we were told that was the delivery date."* (Developer, team S)

Looking at not just the core team members, one can see all the Lake Project teams share the goal of making the distributed agile component work. Although project members faced challenges to achieve this goal, this is actually one of the Lake Project deliverables. BA from team S says "W*hat all of the members were into was this distributed thing needs to work; we had to make it work."* First step was introducing the partnership with the River for team C who were also working on distributed agile context. Separate from that partnering, the Lake Project members are working across of Australia, Sydney, Melbourne (core team) and Queensland (product owner).

The next step was, as part of this project where the agile capability also comes with it, to create a closer relationship and cultural alignment between the teams in Sydney, Melbourne and China. This was achieved through a lot of common functionality and code reviews that are shared between these teams across locations. *"There is a strong effort aligning those norms, values and goals among teams across all locations. I wouldn't say it's been a complete success but it improved greatly."* (Team leader, Lake Project)

There is a mutual adjustment among the project teams regarding expectations so they work towards the same goal and are not working at cross-purposes. There are certain expectations for the Ocean Group partnership with the River to make the distributed agile project component work and be effective. Therefore, the project teams had to put a lot of effort to make sure all the teams across locations are well aligned. They also sought to improve the mutual adjustments among the teams. For example, developer from team S explains that *"Someone picks up a story that's not important but it doesn't happen as much, when the time zones don't overlap and we are not together, it happens more often in that space."*

The project teams start working on that to make sure everyone is very clear on what's important and what's not, because picking up an unimportant story wastes time, especially when they are trying to be ready for delivery on that day. BA from team C explains that practices make the teams more aligned and prevent such challenges *"We kept getting back to our goals again and being very clear about what's mandatory for going live and what's not. So reminding ourselves of the goals and what we are trying to achieve and then the mechanics of how we monitor that to make sure we deliver to the objectives. We also had coaching and feedback sessions."*

This was about getting the project teams to understand the goals and objectives of the project but also how they work as well. The Lake Project agile coach explains, "*At the start of the project, Melbourne and Sydney were more relatively adjusted about the goals and objectives of the project compared with China because we are all the same company. It's more organisational culture that facilitates this adjustment."* Developer from team S adds *"In adjusting the purpose and interests of the teams across geographic boundaries, definitely the most important thing is that you have a shared goal and you want to achieve a shared outcome."*

Another goal that all the project teams commonly agreed on is the goal of delivering to the project technical strategy which is continuously deploying into production. The way the Lake

Project teams deliver needs to be managed the whole way through because their goal is to create a process for continued delivery for whatever they produce to make it easier to deploy. So project teams need to be really efficient when they are coding and keep it in mind that when they code. The Lake Project team leader explains *"When they are coding they need to keep getting back to this and ask themselves 'why am I coding? Is this the efficient way?' So sharing that goal enhanced the team being built to be a very senior and strong team to deliver to project technical strategy."*

The Lake Project teams focus on delivering software to the satisfaction of the business. This is one of the common goals that the Lake Project teams focussed on throughout the whole project and they regularly reflect on how they can improve the coordination and how Retrospectives, Showcases, Stands ups and all the socialising and meetings could work better. This drive towards improvement was one of the notable features found throughout the whole team.

For example, a Showcase setting out the different screens designed was a very crucial point in the project. By showing the screen designs to the business in the Showcase the team received a lot of valuable feedback from the business. The feedback from the business forms the basis of the next action plan. This makes their work more focussed towards business satisfaction and improves their working processes. *"Definitely it's the focus and it was shared between all members across locations, it's what we've been driving towards. It's not kind of just working together but working a lot better together. I guess we work towards the same goal of continuously improving to deliver to business satisfaction and everyone is bringing up this."* (BA, team S)

A lot of goals and values that project teams share in the Ocean Group are already set in their KRAs and get rolled over from one project into the next, hence project members coming from different domains to the project already have those goals in place, thus leading to considerable common shared goals among members. For example, the Lake Project teams commonly aim to deliver a quality well tested product, providing the best customer service and the whole team must leave the build in green state**.** The Lake Project IM explains "*It's our personal development goal to aim for delivering high quality products and that's shared across all stakeholders. They are already in our personal development goals from the Ocean Group and it is transferred to China team as well."*

Shared common assumptions

The concept of distributed agile projects across Australia and overseas was a new concept for the Ocean Group and it all started with the Lake Project. At the very beginning of the project, they were quite distracted in trying to establish a standard way of working and trying to overcome some technical issues. So while focussing on these aspects the Sydney team made assumptions regarding certain story cards which were then not well communicated to China (and visa-versa). Despite being under pressure to complete high priority stories, some of which had been overlooked because of the differing assumptions that had been made between teams, the teams spent time establishing communicating channels and undertaking other activities to align teams in the best possible way and have everyone "on the same page" across geographic boundaries. For example, there were times where there might be miscommunication especially during the beginning stages. They had more frequent catch up sessions with team C to make sure they are aware of the way stories are prioritised. BA from team C explains "*Once we become more fluent we found it easier to communicate. Then, we start to share a common assumption that we will get the information from the BAs and they let us know which one has high priority and all that."* This example indicates that shared assumptions across locations made coordination easier among the teams.

There is also a shared assumption and common understanding between team S, team C and team M that, as a team member, you are expected to either participate in all team activities or challenge them, but you do not have the freedom to simply withdraw from those activities. For example, BA from team S explains that *"If you don't like the way Retrospectives are being run, you are free to challenge that but you are not free to say I am not going to attend. Everybody had the freedom and does say that I would like to be involved and make that better."*

Another shared assumption and understanding is about the way work is managed during iteration or the way the teams manage their work if they run out of planned work in iteration either because a story becomes blocked for any reason or because they have completed everything. In these circumstances, the teams have shared assumptions and understanding of the circumstances and what is allowed or acceptable in all locations to manage them during iteration.

For example, all the Lake Project teams share this assumption that if something comes up in the middle of iteration or in the middle of the day, they cannot throw a card on the wall and say they are doing this. They cannot simply decide during iteration that you are going to work on something outside the team. In these circumstances the shared assumption is to make sure it has been discussed with the lead BA or IM and the decision has been communicated with the team. Team S developer explains that *"They all share this assumption that, this would be spoken about with either the lead BA or the IM just to get a kind of even a verbal confirmation, less because of its approval loop and more knowing that the channel makes sure that the decision and that action is communicated back to the team."* Another case to mention here is when the project teams have got too many blockers on the wall, all the teams know that they need to raise the point, in other words, *"Everyone shares the assumption that someone needs to say it and usually there is someone to raise the hand and say 'hey guys there are blockers on the wall already'."* (BA, team C)

Working agile is part of the job and a project assumption. There was no need to encourage the agility between the Lake Project teams. Team C members are really big agile advocates and very strong in agility approach; team S have members who have experience in agile and that's how they love to approach a project; the same case with team M members. Agile is the way the Lake Project teams work *"We were quite lucky that we didn't have this conversation that 'are we going to do this agile' because the assumption is that this is going to be agile. We did not have really a choice to be honest to do things rather than agile. China team is a company that we are partnering with and they are very famous about their agility approach."* (Agile coach, Lake Project). Indeed *"Nobody thought about it twice, as this is how we work."* (Developer, team S).

One of the Lake Project's core assumptions is not to work on weekends and not to work very late hours. This is shared by all the teams across geographic boundaries and everyone is aware of it. The one about working weekends is interesting. The Lake Project teams work at a maintainable rate and certainly aim to maintain the work and life balance. *"We are going live in November and have plenty of time to manage it. Teams have not worked any weekends and they won't. We focussed to make sure we wouldn't need to do it."* (BA, team S)

There is a shared assumption and understanding by the teams across geographic boundaries that unless someone needs to work hard because they are blocking other members to continue their job, they do not work late hours. Developer from team S explains *"If I'm preventing someone else from working I need to make sure that I complete that piece so they can move on. There is an understanding of getting the job done, so that might mean working 'til a little bit late in the evenings like 7pm but not that late."* As you can see there are shared assumptions among project members across geographic boundaries but at the beginning of the Lake Project, there was a difference in the level of assumptions that the project teams made across locations. The difference resulted from different levels of awareness from context and different working nature.

For example, at the start of the project, context assumptions were stronger in team M and S compared to team C. The team members in Sydney are long term Ocean Group employees, at the beginning of the project they knew a lot about this project compared to people in China that are external consultants. They didn't know the domain, they didn't know the stakeholders. So the reference frame to make the assumptions that they had was totally different and much less developed than people in Sydney, not necessarily because of their location but because they were not Ocean Group employees. However, the differences in assumptions were brought into line through the all teams from all locations meeting in Sydney. During this time they went through extensive workshops to make sure all teams had the same level of understanding from the concept and everyone was on the same page and familiarised with the stakeholders.

In addition, there were assumptions that were only shared between team S and C, not team M. The difference came from differences in the nature of the work in Melbourne compared to that in Sydney and China. The Melbourne team are more mainframe developers, whereas Sydney and China are more front end developers. Some assumptions came from a slightly different reference framework than that of team S and team C. For example, the assumption around ambiguity of requirements; when team S and team C start pieces of work, they tend to be a lot more comfortable with the ambiguity around work compared to team M. BA from team S explains *"We assume that if we have to start a piece not knowing all of the details, we would do best of we can and if that needs to be reworked later that's just something that needs to be planned. We don't have quite that assumption in Melbourne. They tend to assume that if something isn't understood completely before starting then don't start it."*

Application of Collective Sanctions: *actions taken by project team members to condemn the unacceptable behaviour or to reward the acceptable behaviour*

The application of collective sanctions in the Lake Project network is identified through constructs including actions to condemn the unacceptable behaviour and actions to reward the acceptable behaviour (See Appendix C). In summary, I found that the Lake Project teams do not believe in extreme collective sanctions. They condemn behaviours in contrast with their norms, values and goals across all locations. However, this condemnation is not done with extreme or very negative actions, instead they believe in constructive actions and gradual escalation to give the member a chance to behave in accordance with the norms, values and goals before it gets serious. The Lake Project IM states that "*We found gradual escalation an opportunity to clear doubts and misunderstandings before they get to severe sanctions*". They also encourage high performance and achievement through financial and non-financial rewards. I found that both actions to condemn and reward the behaviours reduce behavioural uncertainty by increasing the cost of opportunism (although gradually), decreasing the cost of coordinating and monitoring and providing incentives to monitor inadequate behaviours. The following paragraphs provide more detail.

As mentioned, there is a strong tie among the Lake Project members locally as well as across geographic boundaries. This high level of embeddedness among the teams facilitated the diffusion of misbehaviour of any project member in all locations. Such knowledge about other members makes it possible for the teams to act and condemn the behaviour immediately, which acts as a safeguard to protect the other teams from any unacceptable and unappropriated behaviour.

I found that within the Lake Project team, collective sanctions for unacceptable behaviours across geographic boundaries are imposed on the team members who either are not engaged with the work (low performers) or are too aggressive, use the blame game, are negative, critical and eventually become blockers to progress. The team members across all locations feel safe to raise the behaviours that are in contrast with the project team norms, values and goals and they are usually raised in regular meetings like Retrospectives and daily Stand ups. Through this strategy, the project teams are protected from opportunistic behaviours across geographic boundaries.

The imposed sanctions vary but the general rule is that unacceptable behaviour is dealt with by gradual escalation. This starts with having a conversation with the team member to review the behaviour and find out the reasons for it. Sometimes, the team member will be assigned to a different task, undertake more training or be reported to their HR leader, which may lead to financial consequences.

The study also found that collective sanctions for acceptable behaviours are in the form of rewarding team members through both informal and formal rewarding actions. Informal actions would be words of encouragement and acknowledging and recognising team members for their achievements at regular meetings like Retrospectives and daily Stand ups. The formal actions to be rewarded include financial bonus for high performing team members in their annual performance review or acknowledgement via the 'THANK YOU' system (a mechanism that sends a thank you message through the communication system that is visible to everyone across locations). The following sections set these findings out in more detail.

Actions to condemn the unacceptable behaviour

In the Lake Project, there are parameters for acceptable behaviours. If a team member violates the values, norms and goals of the project, it could be discussed in the Retrospective. In the case of having an unacceptable behaviour from a team member, the usual practice across locations is that the Lake Project IM will talk to the offending team members and outline why their behaviour is not acceptable. After such a conversation team members usually very quickly realise what they are doing wrong. If not, the IM has to follow it up and usually that follow up conversation is quite straight forward, it's more about this is what they've done and this is how it impacted the team. Usually, if the first conversation didn't work, the second conversation would work. The Lake Project IM explains *"So I had a case that there was actually a clash on between two team members and I had to go to the next step and go to team leader. This happened too often and the team leader actually took over the scenario and had one conversation and decided to split the two members and take one to another team because from one conversation he was able to find out they cannot work together."*

The consequences of unacceptable behaviours, as far as ratification is concerned, are more about gradual escalation. The first step is to talk about the issue with the other team members involved, then if it doesn't resolve the issue, the next step is to discuss the issue with the team leader. If the team member feels that the team leader is not supporting in any way, then the next step is to escalate to the higher managers. It is more about whether the behaviour has changed over time. The IM and senior BA will have a talk to the team member and if still the situation doesn't change, then it goes to their HR team leader as a performance issue and escalates further if the behaviour doesn't change.

There is a high level of embeddedness and trust among the Lake Project teams across geographic boundaries and this makes it possible to motivate the good behaviours and condemn the unacceptable behaviours among the teams. For example, a developer explains that *"We had a pair programming between team C and team S. The team C member was not as knowledgeable as the team S member about the context of that piece of work but they both worked very hard together and achieved high quality coding for that difficult piece of work. This high achievement was immediately distributed among all team members across locations because there is a strong bond among us and their high performance was recognised."* This example indicates the strong embeddedness among the Lake Project teams acted as a safeguard to reward or condemn the actions across geographic boundaries.

If an unacceptable behaviour happens that is in contrast with the team social contract and Ocean Group values, team members across geographic boundaries will raise it as a concern through retrospective sessions. In most instances, it will be raised during stand up sessions. This is a joint act from the teams across locations and they feel safe to talk about it openly. For example, the BA from team C explains that *"We had a team S tester [that] was not happy with team C developer, he talked to the colleagues. The team members were agreed he is a real problem and first they tried to work out how to address it together and collaborate about how to solve the problem. In this case, it is resolved at this level, if not they might all go to the project IM and they might all attack the developer, depending how big the problem is."*

The Lake Project IM believes that this reaction also depends on the maturity of the team and how long the teams have been working together. Usually it doesn't happen in the early phases of project because the team doesn't feel comfortable to do that in the first few months, but afterwards teams start to voice these issues. BA from team S explains *"We talked about it and if we need to change the social contract we change it and implement the fines as part of the social contract so if you break the social contract you have to buy cake for the rest of the team or something like that. When we put the social contract together I don't think anyone has broken it since."*

The other scenario of having aggressive behaviour in the Lake Project team is that there were members across locations that were negative or critical during Showcase meetings, Retrospectives or iteration planning. They eventually became blockers to progress. In such scenarios, the Lake Project IM reminds the person of the impact their behaviour could have and that it is outside the social contract that they've agreed to. The Lake Project team leader explains *"There are right ways to challenge things or there are right ways to I guess withdraw yourself form a particular activity if it's not a useful or productive way to spend your time but that aggressiveness or critical behaviour does have this impact on the team and it's not something that can continue."*

In the Lake Project it is important that everybody progresses with their tasks and feels comfortable to voice out their concerns and ask for help as early as possible. This is because not only is the saying *"bad news doesn't get better with age"* (team leader, Lake Project) true but *"sitting on it slows everything down"* (IM, Lake Project) also. To prevent any complications you need to make any issues or concerns are known early and quickly and communicated to as many stakeholders and project managers as possible. Therefore, aggressive behaviours that may intimidate people, such as verbal abuse or very critical comments, are acknowledged by the other team members' subtle avoidance of the perpetrator across locations. Therefore, the project teams are safeguarded and protected against aggressive behaviour that may occur in team discussions or during pair programming across geographic boundaries.

A Business analyst believes practicing pair programming across geographic boundaries is difficult to do due to distance between geographies. He adds *"Pair programming is traditionally a very collaborative and co-located thing. So taking that co-location out of the equation effectively means it makes it more difficult to get that true collaborative sense to it because it does put a barrier between you and it feels like one person is working and one person is watching."* (BA, team S)

The only issue the Lake Project teams had while pair programming was probably the level of engagement and the willingness to really pair program. For example, team S developer explains *"Mostly towards the end of the project I started struggling with pair programming and preferred to work on my own. But the other developer argued with me and insisted that he wanted me to pair with him. I expressed my point during the Stand ups and it was well received by other team members across all locations."* In this example the developer responded by voicing out the issue openly to all the teams to correct inappropriate behaviour in their future exchanges.

Another action to condemn the unacceptable behaviour is that the Lake Project teams jointly deny or prevent the blame game and negativity across locations, "*they kill it on spot!"* (BA, Team C). If the project teams sense that there is some negativity and blaming coming through they try to quickly pass that negativity and focus on how they can fix the problem. *"The blaming game happens very rarely and a lot of team members have been quite good at doing that as well. I found out that as long as there are a few people like that in your team then the whole blaming game tends to die out pretty quickly and it is not acceptable."* (Developer, team S)

Another type of inappropriate behaviour that collective sanctions is used against is a lack of engagement and low performance. In general, the Lake Project manager believes that using negative social sanctions such as exclusion from gathering outside of the office has a negative impact on the project team and it is very unlikely to get a good agile team out of that, that's why it is not a usual practice for them. The Lake Project team leader adds *"I think with the negative social sanctions, you will not get that trust. As soon as you start enforcing sanctions like social activity exclusion, you're effectively admitting to yourself that the trust and the commitment in the team is not enough to get the behaviours that you want and you have to force it by putting such rules and punishment in place."*

The less passionate, or very quiet (i.e., introverted) team members are encouraged gently to get involved. The aim is to enhance the engagement level. For example, if such a lack of involvement occurs during any kind of group discussion, the facilitator of the session often tries to engage with people who are passive by directly asking them for an opinion to make sure that sitting back is not an option. This is a positive way of trying to give active team members an interaction. The Lake Project IM explains *"…we encouraging them gently. So if I'm sitting in a meeting and I know this person knows what we are talking about but not saying anything, I might just go: hey* *what do you think? Do you think it's all right? Just try to teas them out and try to get them involved, if you know what I mean.*"

The roles also rotates locally and overseas, such as the facilitator role for little things like the Stand ups or small ceremonies. The facilitator needs to be slightly more diligent about keeping an eye on people who do not make conversation, keeping an eye on who is not contributing and actively picking up people who aren't engaging. This gives them comfort with the team to do so which they tend to find over the life of the project. This shows them a 'safe-to-fail' environment within the teams and allows people who are normally quiet to engage and get that familiarity. The Lake Project IM explains "*So far [it] has had a very positive impact on their engagement in other activities even if they are not facilitating because they have that comfort. That's a challenge, I wouldn't say that we are all the way there; we can certainly improve on it, but that's the preferred path that the team prefers to take."*

Indeed, the project team takes the responsibility that this does not happen. If there is a Stand up or a Retrospective session and a team member doesn't attend, it is the IM's responsibility to sort that out and the team leader's responsibility to make him or her attend. The Lake Project team leader explains "*…I mean people tend to manage that if we are having any kind of group discussion or whoever happens to be facilitating that, often try to engage directly with people who are passive to make sure that those sort of just not sitting back and let all happen around is not an option.*" For example, in one instance, a tester approached another team member directly about his level of engagement and suggested that the other person should contribute with more information, ideas and helping behaviour in the team meetings.

When the Lake Project team members are not performing well, the first action is to have a conversation with the person and try to resolve the issue at the first level before escalating it to the next level. There are lots of discussion points in the conversation; they try to work out what the problem is and how to resolve it. For example, if someone is not responding maybe there is a gap in their skill set and coaching might be required. In such cases, although the Lake Project team leader is accountable for working through this issue and solving the problem, he is also responsible to provide feedback about the existing gaps to the team member's HR team leader.

The Lake Project manager explains *"The Lake Project team, like any other agile team, is a team where they try to pull each other up, especially with a high performance team that does not like* *the not contributing members and don't have much tolerance for that."* They try to create a culture where people are comfortable and they can pull up by high performers ideally. If that does not work and if there is bad behaviour that does not get resolved then there will be a discussion at the team level. If it still remains the same then they will escalate the issue to the formal performance manager. Bu the goal is to resolve it within the team level.

One of the strategies to manage low performance in a particular task is to rotate the assigned task and to make sure that particular type of task doesn't tend to get assigned to that person in future. One of the reasons for low performance in a particular task could be lack of interest. A specific example of this would be when, although the Lake Project team is a self-managing team who choose the tasks they are interested in, it happened that a team C user interface developer was low performing on a task which was not very well aligned with his interests. BA team C explains *"We had a chat and found out that due to his personality he was not able to saying no to the task even though he was not interested. We then rotate the task and made sure he will get the tasks in his areas of interest."*

If the case is not resolved at the team level, in extreme cases the team member has to be removed from the project. For example, there was an instance where a team C member wasn't listening to instructions and this type of behaviour was repeated. He missed couple of deadlines and it became obvious that some changes needed to happen. They had to balance what is the risk of removing his capability and his knowledge, or the risk of having his disruptions in the project. BA from team S explains that "*We tried to resolve the problem through talking with the person as the first step but to no effects. The case was escalated and finally we had to remove him from the project completely even though he was an excellent team member technically. Because he was very disruptive it definitely influenced team members around him performance wise."* In this example the team member was excluded from the project in response to his unacceptable approach and behaviour.

Another approach to low performance is a combination of trying to give that person slightly more direct responsibility but also more support to deliver the work assigned to them. "*This means making sure few other people in the team are aware of the issue and are actively engaged in and trying to support the person trying to get through that thing."*(Team leader, Lake Project)

In this way they try to determine whether or not the low performance is a question of capability or attitude.

Beside the non-financial consequences, there are also financial consequences. For example, if talking through the problem does not work it will be escalated to the HR leader which sometimes leads to have financial consequences. Because the project team members are on a bonus system, they have to have a minimum success and performance level in the annual performance review. If the project team members do not get to their successful performance level, they will not get their bonus and any salary increases or promotions.

One of the joint actions by the Lake Project teams across geographic boundaries to condemn the inappropriate behaviour is avoiding the blame game. This act has been discussed as a shared common norm across locations before. If a Lake Project team member raises an issue, the team member cannot be blamed and insulated for the raised issue. If any team member tries to behave in a blaming fashion, all the other project team members jointly condemn the action and do not let this happen. Instead they try to focus on resolving the raised issue and not waste time on blaming. Finally, the Lake Project teams collectively condemn any sort of negative attitudes, comments or behaviours. On the other hand, the positive behaviours like high performance and achievements are mentioned and circulated across locations among the teams to reward the positive behaviour.

Actions to reward the acceptable behaviour

The Lake Project team members are rewarded for their acceptable behaviours through informal actions such as words of encouragement and a show of recognition and also through formal actions like financial bonus in their performance review and the 'THANK YOU 'system across locations.

Informal rewarding actions:

Due to the distributed nature of the Lake Project, in most cases the teams across geographic boundaries are rewarded through words of encouragement and addressing their achievements during Stand ups or Retrospectives where majority of the project team members are gathered. BA from team C explains *"We reward members simply by clapping when a story is done or if* *someone achieves something that they worked hard on, they will get acknowledgement during the Stand up."*

The rewarding and appreciation of the team achievement by senior managers was very visible when the project went live. During this appreciation time, they had the other teams (team C and team M) connect live with team S in Sydney so the whole team could be recognised for the outstanding job they have done. For example, the Lake Project IM explains *"I know when we went live, the product owner and the other program manager actually came down on the morning of go live to say thank you personally, which was great."* They send thank you messages to other teams across locations through Skype camera. The team S developer adds *"Another team leader had come across and sat with us and shook our hands and said thank you. It was a nice gesture."*

There are periodic rewarding activities for the whole team like all going out to lunch together and the project pays for that. There would never be a scenario where somebody on the team would be excluded from that for any reason, especially not for performance. This rewarding action is done across geographic boundaries in China and Melbourne where team C and team M are located as well.

In the Lake Project team, it is so much more about the informal rewarding actions for high performance, besides the formal rewarding systems such as financial bonus and the THANK YOU system. It's more about trying to clap those things that are really positive examples of the kind of behaviour that the team expects. For example, the Lake Project team added a standard item in Retrospective meetings where the team members post 'Thank You' on screen to highlight and show appreciation for the other team members that have gone the extra mile. For example BA from team C explains *"Basically a team member would say 'I would like to thank this other team member for doing this during the last iteration which really helped me out' and that might be for staying late to get one particular piece of work done because it might have been taking time out from their normal work."* The motive behind this rewarding action is to push good examples out there and hopefully achieve a flow on effect where people are motivated to adopt those behaviours and to engage more of those behaviours within the team because it is regarded as a very positive action by the team and something that the team would like to see more.

One of the informal rewarding actions for high performers is to talk highly about them to other peers whether they are in the project team or not. In this way they distribute the good words about them. Then the members are rewarded reputation wise, gain respect and everyone recognises them as high performer and committed team member. They can also get awarded more advanced tasks and get involved with more decision-making activities.

Formal rewarding actions:

There is a 'THANK YOU 'system across locations where you can send a thank you and it will be broadcasted at the department meetings. The departments get together once a month, compile them all and then socialise it at the departments. It is accessible to all the Lake Project team members and everybody can use it, it's an online system to send the thank you message. In case of team S and team C, the message gets triggered to the person's HR team leader at the end of the month, and in case of team C, it is also communicated to their China HR manager for their recognition.

However, not all of the project team members feel comfortable about this formal rewarding system. For example, BA from team S believes although it's an appreciation and recognition, strangely it can be embarrassing sometimes. *"Sometimes team members are shy and do not feel comfortable to be rewarded in this way because there are about 150 people at the department and they are linked by video conference all over the Ocean Group branches Australia wide (Sydney, Melbourne and Queensland) plus China"*

Another formal rewarding system for the Lake Project team members is a financial bonus at the HR level. It is the IM's responsibility to make sure that HR team leaders (in the case of team S and team M) and the HR manager (in case of team C) get the feedback about particular team members with high performance, the outstanding job they have done and how they have done it. Then the team leaders get that information and reflect it in their performance review. There is an annual review cycle and rewards come through bonuses, promotion and pay rises. Similar rewarding systems apply to team C as well.

Application of Reputation: *the perception that project members hold about another member's character, knowledge and skills across the project teams.*

The application of reputation in the Lake Project network is identified through constructs including social expectations and individual expectations. Social expectation is defined as the extent to which the project team members' expectations are important to the other team members and individual expectations are defined as the extent to which the individual project team members' expectations are important to themselves (See Appendix C).

I found that not only are the project team members' social expectations from the other team members important, but the team members consider it important to fulfil their obligations towards the other team members across geographic boundaries. I found this was stronger in team C in comparison with Australian teams. Therefore, it is important for the Lake Project team members to have a positive reputation and maintain it. Having positive reputation increased trust of the other team members and helped them to make better decisions. This protects the team members across locations from any behavioural uncertainties. The following paragraphs provide more detail.

I found it is important for the Lake Project team members to fulfil the social expectations of the other team members across geographic boundaries. Factors that motivated team members to achieve this include agile iterations work base, high visibility of the teams' activities and finally the importance of delivering to the satisfaction of the business due to the Lake Project being a high profile project. However, achieving social expectations was felt more strongly and critically in China because team M and team S are customers of team C. Therefore, China team members put additional effort to prove themselves and gain positive reputation.

The same case applies to individual expectations. I found that there is a strong tendency among all the Lake Project team members across locations to gain other team members' trust and be considered as a competent and reliable professional. However, the need to achieve such positive reputation was stronger in team C for the same reason, that the Ocean Group team were considered a customer to team C. This was a strong motivation for them to gain positive reputation and image. I found that overall the Lake Project team is regarded as having a highly positive reputation team in all the Ocean Group branches Australian wide. *"It's one of the teams that is known company-wide to have succeeded in a number of areas, especially in distributed agile and we had visitors from all over the place to see how we are doing it and succeed in it."* (Project manager, Lake Project)

Because the Lake Project is such a high profile project, there is a lot of visibility amongst everyone in the business as to where they are, how they are performing and what the team is like in general. The Lake Project team members are well aware of that and wanted to have a positive reputation as a whole team. Another reason for having a really strong reputation across locations in the Lake Project is that majority of the team members are chosen for their reputation in delivering and their strong skill sets. That is why it is very important for them to maintain their positive social and individual reputations within the team and in the wider network as well.

I also found that although the Lake Project team faced a lot of challenges due to the agility plus distribution nature of the project, they were able to get through them to the extent that they became a reference point for other project teams. The project IM remembers *"There were times we felt the distributed way of working is really not working out for us. We had too many issues with technology but we were able to work them out. Now people keep coming to us to ask us how we did it, it gives us the feeling of being proud of the reputation we earned, it gives us motivation to go to next step and keep improving."* (IM, Lake Project)

One fact that helped the Lake Project teams to be able to build a positive reputation and have that trust between them in the early stages of the project was that they were very well aware that everybody chosen for this project is a capable person. The reason for assembling such skilful people for this particular project, was that this project was very important for the Ocean Group Company as the first distributed agile project. *"Even if I've never had a word with them there was always that sort of feel at the beginning to make good judgement. So if I've never worked with someone before I know about their reputation from others that worked with them before."* (Developer, team S) This example indicates an existence of initial trust among the teams helped to make a positive reputation environment within the team.

Indeed, there is a lot of trust in team members' ability to perform the tasks they have been allocated to across all locations, but it can vary based on the team members' relationship and how long they have worked with each other. For example everyone from the Ocean Group knows each other and have worked together for longer than they worked with the China team but it does not mean that there is always higher trust among the Ocean Group members compared to China team. The Lake Project team leader explains *"There is probably more trust in* *the Ocean Group teams but it's a relative statement. There is still high trust with team C also. It's all about their track record [reputation], trust comes from track record. We found the China team's behaviour consistent against the tasks we asked them to do since the beginning. They show high ability to perform the tasks allocated to them".* This example indicates that building a positive reputation along the way with the other project team members allows others to trust them more in their ability to perform the tasks, so trust is part of gaining a positive reputation.

One example of trust based interaction across the team was when team C and team S had to integrate a big part of the project through the backend and they were not able to provide any visibility to team C for varied reasons. Because there was a lot of trust across the teams, team C followed team S. That was seen as a very good reputation builder for the project team because they had no previous examples of this case before that they do that piece of work using partners. Therefore, the Lake Project team members had to trust it would work and they were able to do that work in the Ocean Group using the partner company. The project teams were paired in China and Sydney to do that work but if there was no trust it would have been impossible to do the integration to the backend. This example indicates reputation between the teams across geographic boundaries is based on trust.

On the other hand, I found that although there was a strong sense of having a positive reputation across locations, the Lake Project teams faced a few reputation challenges as well. For example there were few stakeholders who had the reputation of being particularly tough and strong headed in questioning why certain parts of the project were late. The project team members either experienced it themselves or heard from other team members. That made the project team members be more cautious about what they say and how things are communicated to them. "*If I have been approached by them I was a lot more cautious about what I say and about the issues and challenges. I rephrase them in different matter. So it did make the coordination a bit more challenging, it was not that easy."* (Developer, team S) In addition, I found that the existing embeddedness among the Lake Project members across locations enhanced the flow of effective information about the teams' behaviour in all locations. This facilitated the teams to be well informed about other teams' reputations and trustworthiness. The following sections set these findings out in more detail.

Social expectations

Social expectations of the team members from the other team members do exist and are important. One of these expectations that is very well fulfilled by the project team members is that they voice out any issues or difficulties immediately. This always gives them a positive reputation because they are seeking help and support. *"There is always the expectation that if somebody is having difficulties, then he stands up and talks about the issues he is facing. This becomes one of our usual practices now and gives us that feeling of trust in our interactions."* (BA, team C)

Another example is the people in the team that organise social events for the project team members. There is always the expectation to have some social event every couple of weeks and this generates a very positive reputation for the organiser. The organisers manage this responsibility very well and the team members are very happy about it. BA from team S explains *"Social event organisers do their job and fulfil the team members' expectation really well, which is why they are respected and appreciated for their role."*

One reason that the project team members achieve their social expectations is the flexibility that is supported by agile and it's the same case in all three locations. Agile helped the project teams to be consistent when fulfilling the expectations because they have iterations and they are committed to have them done with a period of time. The Lake Project IM explains *"We are all experienced and when we say we are going to get something done by a certain time or day, there is that social expectation that we to do it. Especially because we are working in an agile environment the expectation needs to be fulfilled during that iteration."*

Another agile practice that supports fulfilling social expectations in the Lake Project is the high visibility of activities that agile bring to the project, especially for distributed teams this is very critical. Because the other project team members across all locations have high visibility of the team members' actions and their activities are very transparent to others, they do not have a chance to hide their activities. This motivates project team members to present a positive reputation towards other members as much as possible and fulfil expectations of the other project team members. In other words, existence of such visibility and transparency of teams' activities makes fulfilling social expectations and individual expectations very important. Developer from team S explains *"Your progress is regularly monitored and visible in agile and in every Stand up. Then someone else has to follow up that task in next step so you want to make* *sure that your part of job and your stack is done as best as possible. So having that visibility and transparency amongst all your team members, as a developer I used to have more motivation to try."*

As mentioned before, most of the Lake Project team members are assembled based on their delivery and skilful reputation due to the sensitivity and importance that this project had for the Ocean Group Company. Therefore, majority of the project team members are knowledgeable team members and knowing that someone has a particular skill set and a particular reputation makes the team members expect that person to help and support them and move things along quicker. The Lake Project team leader explains *"You have your people that are very strong tech leads and people feel free to go to them and expect to have an answer to what they want from them in the first place. That's what we respect that person for, that quality, and when we built the team as well that's what we looked out for."* This example indicates in order to gain a positive reputation and be more involved, it is important that the team members fulfil their responsibilities and show their skills and knowledge to the other team members.

For the Lake Project team it is important to deliver to the satisfaction of the business in order to gain a positive reputation, not just from the business point of view but also among the other IT teams within the larger network. However, this importance has a different priority for team C compared to team S and team M. The River (team C) is an IT consulting company and the Ocean Group Company (team S and team M) is their customer.

This partnering arrangement makes gaining a positive reputation and fulfilling social expectations more critical and important for the China team. The China team not only needs to get the software out and make sure that it is represented in public but on top of that they have to prove that they are valuable to the Ocean Group because they are their customer. *"You get the difference between the consultant [team C] and someone who is permanent in the company. We are all very hard working but there is more eagerness for the consultant to impress reputation wise. So I think the reputation was stronger on their side because they were in a different position."* (Team leader, Lake Project)

For example, one of the practices used in team C to gain positive reputation was to be more precise on the quality of work they deliver compared to other the Ocean Group teams (team S and team M). BA from team S explains *"In Sydney we do pairing and part of the pairing is code* *review but in China they do an additional code review to the pairing. That could be because they try to make a positive reputation and provide a higher quality task."* This example indicates that earning positive reputation and fulfilling social expectations had higher priority for team C due to their position in the project.

The Lake Project team faced a number of challenging situations at the beginning of the project but they were able to master the situations through successful delivery which created enhanced trust within the Ocean Group and the business. The Lake Project was the first distributed agile project for the Ocean Group network and all eyes were on them. They had an issue with someone who tried to damage the team's reputation and ability by regarding them verbally as a poor quality code building team. They managed the issue by acknowledging the issue and making the quality of the job they were doing visible. The issue was that people could not see the quality of their product so they needed to make visible the quality of their product. They developed a chart which is very specifically lists all of the issues that were happening in the product and distributed that around. They also use a large piece of wall where they outline all of their issues on the cards and split them out into different sections.

In this way they were able to make it very visible to everyone that this team is managing the quality of the product very well. The Lake Project manager remembers: *"We have demonstrated we can deliver high quality code and we have shown that we are committed."* As results emerged, the business representative started to see and, importantly, to describe the team to others as highly skilful and committed. This example indicates how important the social expectation for the Lake Project team members is. They immediately start to respond and fight back their reputation and show others that they are fulfilling their responsibilities very well and as expected.

There is an expectation that team members expect the other team members to fulfil their responsibilities because it helps the team get along. *"Things like 'I just won't finish something because I can't be bothered' that behaviour is not really tolerated, it's a harsh word."* (BA, team S) Although the team members feel responsible and committed towards the other team members' expectation to perform to their role, there is also an understanding that there are skill level differences within the team. So it comes easier for some and not others just because of different experiences in different projects that they worked on. *"So beside the expectations* *there is also 'look if you've done this before and I haven't, we'll help each other through this.' "* (IM, Lake Project)

Another point to mention here is that although there is social expectation for the team members to fulfil their tasks, this expectation varies for different team members and it is based on their reputation. The IM explains that they take in the reliability and character of the other team members before interacting. *"Everyone is switched on and gets things done but the expectations are different. For example, you know from previous experience if someone tells you I'll get this done by tomorrow, that they either probably don't get it done by tomorrow but they finally deliver, or from previous experience you understand that they always delivered on their word and they will deliver by tomorrow and you can rely on that."* (BA, team C) This example indicates that although there is the expectation from the team members to fulfil their tasks, it depends on their reputation. Sometimes even if there is a delay it is acceptable because they have showed they will always complete the task but it might be slightly later.

Although in majority of cases, the Lake Project team members fulfil their obligations towards the other team members' expectations, there have been cases that they did not fulfil the other team members' expectations and did not act in line with strategy or in line with respecting the other team members. For example, as a distributed team, the Lake Project IM and the team leader expected all of the team members across all three locations to come to him in the first place for any feedback or issues. However, there have been scenarios where the team members did not go through HR escalation as was expected from them, instead they passed the IM and the team leader and spoke directly to the program manager at program level. *"I was informed about this by the program manager. Such actions make a mess in communication and create a huge issue for yourself because you cannot focus on other matters, you have to focus on the matter you have at hand. This complicates the coordination."* (IM, Lake Project)

In order to fix the scenario the IM had to work for days and explain the situation to the program manager and other stakeholders. He also spoke to the whole team in all locations about the need to follow the escalation points for any issues. *"Because obviously the program is paying for this project, they become concerned if Lake Project team members aren't working together. It really impacts on the reputation of the team as well, and it gives a negative impression to outsiders."* (Team leader, Lake Project) This example indicates the importance of fulfilling social expectations by the project team members. It also shows that the sensitivity of the issue can not only damage the reputation of the team members but the whole team.

Individual expectations

The Lake Project team members consider it important to fulfil their obligations towards the other team members locally and across locations in order to keep their positive reputation. The Lake Project team leader explains "*There is a big sense of team and team members do think it's very important not to let other team members down. Not only for the reputation of the company and partners that are involved but I think it's about keeping the team together ... it's a personal obligation that everybody feels."* For example, BAs in Sydney discussed and decided about the way they started to write the stories, but BAs in China requested them to consider a different way to make it easier for them, and the Sydney BAs accepted it. *"We did it, it wasn't a big deal for us to adjust, if it meant that it helped them understand and make their life a little bit easier, then that was something we could do for them. It was important to keep the team working sense within the team."* (BA, team S)

All the teams across locations attempt to make a positive individual reputation but this sense was stronger in team C in China than Australian based teams (team S and team M). Given that China is 100% partnering with the team members and there were no permanent staff, it was important for them to show a positive image to their client (Ocean Group). For example they did extra activities at night from about 6pm to 8pm, like learning about the industry or applications and they made it visible to the Ocean Group. *"They did it because it was important to them what our perception is of them."* (Team leader, Lake Project) They always tried to prove themselves and make sure that they are a valuable asset to the team. "*We had people in China that used to pick up tasks as soon as they came in, like pick up three to four tasks and work on them as much as they could to get them done as soon as possible. While we don't encourage picking up multiple tasks they persisted at doing so. So it's a way of trying to prove themselves."* (IM, Lake Project)

In the Ocean Group network, it is important for the individual team members to demonstrate their skill level and knowledge to earn a positive reputation. Individuals with a positive reputation will become more involved in the team (e.g., in the team discussions and decision-making activities) and are rewarded with more advanced tasks. A developer from team S explains: *"I think once you do help someone enough to let them get to know what you're capable* *of, they become aware of your skill level, they will naturally trust you a little bit more and involve you more because they are satisfied with what you've been able to offer to them."* Once a team member has gained a positive reputation, the information spreads through the network via *"word of mouth*" (BA, team S).

One point to consider here is that team members need to understand what other team members' expectations are from their role in order to be able to fulfil them properly and keep all stakeholders and other team members happy. Otherwise they never know who the responsibility lies with and they have issues that are not addressed in a timely manner because those expectations are not matched. They might have a different perspective of what the expectation is from them. For example it's important for a BA to understand the business expectation from their role. The Lake Project IM explains "*A team member tells me I don't have access to this and they might have a different expectation that I'll go and get them access. While I probably ignore that email because anyone can get access why should I give you access. And then that issue just sits there."* This example indicates it's important that expectations are understood by the team members so the other team members can rely on them.

There is a strong sense between the project team members of wanting to be considered competent, professional, reliable and trustworthy by the other team members because they want to win other team members' trust. This is very visible; they are always seeking to get as much information as possible about the context and background and want to know why they are doing things in order to help them to make better decisions. For example, positive or negative reputation can be pointed out easily by the other project team members due to the agile visibility that exists in the project and the team members have to give an update of the story card that they are working on, on a daily basis. Because there is awareness and other team members know what you are working on and how long you are spending on that part at any given time you have a strong tendency to make a positive impression on other team members. *"As a developer I had more motivation to make sure that I do deliver that work on time and if there is any issues, raise that issue fast."* (Developer, team S)

From the reputation point of view when a team member has a strong positive personal reputation he can get things moving and get his suggestions through a lot quicker because everyone trusts him and everyone like to follow him. A business analyst explains "*So when [we]* *have pairing across countries, [we] found the first person to pick it up was a tech lead and everyone followed that person because everyone looked up to that person as one of the stronger people in the team. It's definitely the same case for all three locations."* (BA, team S) Another reason for individual team members to have a positive reputation is because it effects how people interact and communicate with them. *"If I'm doing a job wrong they are going to micro manage and question my ability which is not much fun."* (IM, Lake Project)

Application of Coordination: *the ease with which interactions between project team members are conducted across project teams*

The application of collective sanctions in the Lake Project network are identified through constructs including ease of work with the other project teams and availability of communication tools (See Appendix C). I found that coordination is eased and enhanced through practices and tools provided to the Lake Project teams across geographic boundaries. The following paragraphs provide more detail. The Lake Project team was, for a long time, a distributed agile team; every single team member had to be on top of it all the time and be transparent as much as possible otherwise the communication would easily break down.

I found that coordination is enhanced through different practices including existing high transparency and visibility of project teams' activities, co-locating distributed teams from all locations at the beginning of the project (as well as throughout the project), coordinating roles in the project and finally, a high level of trust among the teams across locations. High visibility of teams' activities was possible through an agile electronic wall called JIRA. This made it possible for any team member to have a clear idea about other team members' activities and enhanced the coordination among them, especially across locations where the physical visibility is limited.

In addition, coordination is facilitated through technology and availability of communication tools for all the project teams across geographic boundaries. This made it possible to overcome a lot communication and coordination challenges involved in DD projects. For example, the Lake Project teams have access to smart boards, Skype, Lync (a messaging tool specifically designed for the corporate and a communication solution built by Microsoft), JIRA, teleconferencing and screen sharing to use as communication channels across locations. Availability of these communication tools facilitated the coordination among the teams in general and the teams across locations in particular.

The Lake Project team members assembled from different domains in the Ocean Group and the River, the partner company. This allowed different team members to bring different backgrounds, ideas, suggestions and different ways of doing things to the project. For example a developer from team S believes "*We benefited from that because a lot of people bring a lot of knowledge and practices from their previous domain and previous projects that they worked on, into the Lake Project. It helped more than anything else as we took advantage of different backgrounds."* This example indicates that different backgrounds not only did not make the coordination challenging, but they also eased working and communication among the teams.

I also found that beside difference in the team members' background, they also benefit through some coordination of identified roles in the project. For example, technical lead (tech lead) is responsible for providing coordination among developers. The tech lead is responsible for making sure that the developers do not work at cross purposes; they are on a single page and on track. Such coordination is also done through every Stand up, making sure every member is working on the same page. Half of the tech lead role is about coordinating the developers and the other half is about making crucial technical decisions.

As discussed previously, there is high level of trust and competency among the project team members across geographic boundaries. I found that the existence of such a trusting environment encouraged the team members to voice issues as early as possible to avoid more complications in later stages, thus making communication less challenging across locations and enhancing the coordination among the team members. Finally, I found time differences and public holidays sometimes complicated the coordination during the project but it was nothing to do with the way project team members behaved. The following sections set these findings out in more detail.

Ease of work with other project team members

Ease of work with the other team members in the Lake Project was facilitated using practices including co-locating teams and transparency in the project team members' activities across geographic boundaries.

The Lake Project was a large project, but co-locating the teams from all locations for a couple of weeks at the very early stages of the project, and also repeating this practice through project life, meant many things in the Lake Project could be possible that otherwise would not. They were able to exchange and share knowledge and experience, learn the process and how things are done on the other sites, build face to face relationships and pick up the problems much quicker. For example, one of the team S members went to China and attended the Stand- up meetings. By being there and experiencing the meeting as the locals do, he was able to pick up the issue of not having any meetings at lunch time. *"It was something that we cannot do from here and the team does not raise it because it is not obvious to them. That is why both ways of exchange is important."* (Agile coach, Lake Project) It is very important to frequently repeat this co-locating practice throughout the project life and give the opportunity of being visited to all team members and all locations.

Such practices also helped the whole team to make sure they were all on the same page when they started, which is very important. If you build the strong ties among the team members, the availability of the context information helps to improve the coordination among the team members across geographic boundaries.

Another example that made the effectiveness of this practice even more obvious was from a Sydney team ex-developer who became IM of another distributed stream at the end of the project. She believes not having such practice at the beginning of her new project made it much more challenging for them to communicate. *"I believe if we had the co-locating practice at the beginning of this project we wouldn't be struggling with the context. I am talking now to see how we can do it for this project as well."* (Developer, team S)

One of the main tools that improved transparency in the Lake Project team members' activities was using the agile wall across all locations. The agile wall gives all the team members visibility on what tasks they need to work on for that given iteration, what work is in progress and what work is done. There are two versions of the agile wall in Sydney and China, a physical wall and an electronic wall called JIRA. The team members in Melbourne preferred to use JIRA only due to nature of the work they had (mainframe) and they thought it's kind of a wall maintenance exercise between China and Sydney.

Everyone in the project team has access to JIRA and is very much visible to all project stakeholders, even the product owner. JIRA provides the team members with a sort of visual representation of tasks. The same sort of wall is physically there to give a Kanban view of the wall. Kanban wall is a wall where the project team members can see what the tasks are and what status they are in, the same version of the physical wall exists in JIRA that is used during the Stand ups. The team members talk through the wall during each Stand up. It helped the teams who had the physical wall at their site to be able to move the story cards over. However, the physical wall involves a lot of maintenance because you need to make sure that the tasks are moving properly and whatever happens needs to be seen in the soft copy because the Melbourne guys relied on the soft copy. *"There were actually three walls to maintain but we gained a lot of benefits from the walls. We had a single person allocated responsibility for the walls' maintenance and to make sure that JIRA wall is synchronised with the physical walls."* (BA, team S)

In addition, while the project team members have screen sharing sessions across locations and talk about the works in progress, they all have the visibility through JIRA. At any given time during the day, anyone can come in, log in and view the wall and get to know what the status of each task that is in progress. The Lake Project team leader explains how heavily they rely on JIRA for transparency in their activities *"Developers mark the task they are working on currently and once they are done they place it into the testing column and that's when the testers know this is the work that they need to look at and are working on today. Once they have gone and tested everything that was on the testing column, they will move those stories or tasks into the user test column and then the business gets to know that they need to get the user acceptance test done on that."*

The whole process of maintaining the JIRA and physical wall synchronicity was well worth it because it made the coordination among the team members across locations much less complicated and they were able to communicate much easier by having visibility of project task status. This example indicates the visibility of activities that was made possible through agile walls, both the physical and soft version, made the communication and coordination between the project team members much easier across locations.

Availability of communication tools

The Lake Project team members have their normal supporting processes, which becomes possible via technology. There are a lot of communication tools available to the Lake Project team members across geographic boundaries that they can use in their exchanges. The project team members use these tools and technologies to provide better communication channels for the team members across locations. They use any possible technology and communication tool to make the distance less apparent and make work more transparent and visible across locations. This is a very critical practice for them due to the distributed nature of the project. For example, they communicate through smart boards, Skype, MOC, Lync, JIRA, teleconferencing or sharing desktops (screen sharing) while they are pairing across locations. These tools are readily available to them. A developer explains *"I usually pull up JIRA and I screen share my screen with everyone who is on teleconferencing. Developers use the screen share when they are pairing."* (Developer, team S)

Lync used to be called MOC, but the Lake Project teams moved from MOC to Lync as the preferred communication tool for screen sharing. The Lake Project IM explains that *"At the start that when we built this team, there was the phone obviously; there was also face to face communication using Skype call. We visually saw each other through Skype during the Stand ups and used our JIRA wall to get the updates, we screen share using a software program called Lync."*

Most of agile work practice communications such as Showcase meetings, Stand ups, Retrospectives, IPM (Iteration Planning Meeting) and tech catch ups are done through Skype calls. A daily scrum of scrums was done through teleconferencing. The project teams' aim by using these technologies and communication tools was to collaborate across locations through face to face communication as much as possible.

Application of Safeguarding: *the degree to which interactions between project members are protected across project teams*

The application of safeguarding in the Lake Project network is identified through constructs, including rules and procedures to deal with problems, the amount of monitoring, equity among the teams and trust-based interactions (See Appendix C). In conclusion, I found that opportunistic behaviour in the Lake Project team is very low due to the practices they use to safeguard the transactions across locations. The following paragraphs provide more detail.

In order to be updated with rules and procedures, the Ocean Group employees were required to go through annual compliance training and pass a test to make sure that they are knowledgeable enough about rules and procedures to deal with problems and unacceptable behaviours. Similar systems also exists for the River employees.

I also found safeguarding of the Lake Project teams was enhanced through different practices by all the project teams across locations. These practices include using success sliders as reference point, a gradual escalation system to deal with inappropriate behaviour (although the intention is to resolve the problem at the team level), discouraging negative comments and behaviours, resourcing the Lake Project with team members that hold strong organisation and agile values, increase transparency and visibility of the teams' activities across geographic boundaries with the aim of reducing amount of monitoring required, regular sanity checks and creating a trusting and 'safe-to-fail' environment to voice problems as early as possible.

Maintaining the fairness and equity among the Lake Project members across locations is another indicator to protect the teams from opportunistic behaviours. I found that although there have been some challenges throughout the project, the intention was always to keep the balance of opportunities for all the project teams across geographic boundaries. If a team member had been given an opportunity that is very interesting, they made sure the other team members get the same opportunity as well.

As mentioned before, one of the major practices I found that facilitated safeguarding was creating a 'safe-to-fail' environment in the Lake Project team. I found there is high level of trust among the project team across geographic boundaries. I also found that honesty and trustworthiness were two main values that facilitated a 'safe-to-fail' environment for the project teams across geographic boundaries. Indeed, the Lake Project teams believe in celebrating the failure instead of spreading negativity. The safe and trusting environment made communicating among the teams much easier and quicker with less amounts of monitoring required. These values are embedded in the Ocean Group and the River's values and the teams are frequently reminded of them. Such values motivate the teams across locations to easily and openly talk about the issues, their failures and wrong behaviour.

For example, during one of the Stand ups a developer in team C voiced out and said he is sorry but he really screwed something up. Everyone reacted in the manner of 'let's correct it and let's do not do that again'. The Lake Project IM described the environment as *"Very 'safe-to-fail' – let's celebrate failure and make sure that they are not locked in and isolated."* The project teams across geographic boundaries are constantly encouraged to raise issues in the team's regular catch up sessions and meetings such as Retrospectives and Stand ups. This example indicates how a trusting environment encourages the project teams across locations to raise problems as early as possible to avoid complications and behavioural uncertainty in the future. It also indicates that having a 'safe-to-fail' environment facilitates safeguarding by reducing monitoring cost. The following sections set these findings out in more detail.

Rules and procedures (or strategies) to deal with problems

As discussed in previous sections, project success sliders are clear indications of the project goals and business and the stakeholders' expectations. Having such indications at hand in the Lake Project made dealing with issues and problem much easier though the project life. The Lake Project manager reports "*The sliders create a context within the project that enables project team members to make decisions throughout the project and will assist in moving forward with decision making in the event of a problem or conflict.*"

The Lake Project uses a gradual escalation system to deal with issues such as low performance, not involving the team members or any inappropriate behaviour that affects the productivity of the team. It all starts with a verbal conversation with the project IM, if the behaviour continues then the team leader will get involved and if the behaviour still doesn't correct then it will get more formal and escalate to the team member's HR team leader. If it's affecting his ability to perform his job or the team members around them then it will go to performance management. There, they write down and identify actions then track those actions; if it gets more serious then it goes into first warning, second warning and then on to being managed out. Therefore, if there is a conflict between the project team members there are HR procedures, but they only get involved when it gets serious and when it cannot be resolved by a conversation within the team.

For example, there was a case where developers had a very light-hearted conflict during pair programming. A young team member of team C was requesting constantly to pair with a developer in team S because they were both working on the same piece of work. The team S developer broke it down and gave each team member a piece to work on their own but still the team C developer requested to pair it all together. Developer team S refers to this and explains

"We resolved the conflict through reasoning, expressing, sitting down and getting the right people get involved and just talking it through." Finally they decided to spend an hour working on their own and then they start pairing and work on piece of work together.

The intent is to resolve the issues at the team level as much as possible before it gets escalated higher. The Lake Project team leader explains *"sometimes the issue is just that someone doesn't understand where the other person is coming from or even a different language, so if something like this occurs they sort it out around a conversation."*

The Ocean Group has its own rules and practices to deal with unacceptable behaviours. The project teams get updated about these rules and practices and what is expected of them annually through their annual compliance training. This is obligatory for all the Ocean Group employees. After they study the latest rules they have to pass a test. If the employees do not do this test on time they do not get access to the internet because they have not agreed with compliance by passing the test. There is similar system in the River for the team in China, where they are updated with rules and procedures through their compliance training - with a small difference: they do a small test at the end of the training on the day itself.

One of the practices that the Lake Project uses to deal with problems is providing a trusted environment for the teams across locations. Trust is an organisational and a project value that has been transferred strongly to the project teams across geographic boundaries. This trusted and open environment encourages the project teams to talk about any issues at the very early stages and not allow certain behaviours to develop as a result of hiding information or dealing with the stress of any issues or risks that have been hidden for a while. This pushes out the unacceptable behaviours and opportunistic activities.

The other practice used by the Lake Project teams is that general negative behaviour is not encouraged or bred by anyone. If there is a negative comment spoken about any subject by a project team member, all the other team members ignore and pass the negative comment and do not let it grow. This procedure prevents any blame game in the project and it seems to be a sort of mindset and culture for teams across locations. The Lake Project team leader explains that *"The project has been very blame free, well we have been the type of team where we were into 'where is the problem, let's solve it and not waste time blaming people, let's just go ahead* *and fix it, because it is happened'."* This practice reduces the likelihood of any opportunistic or inappropriate behaviour in future.

One practice that was used to reduce the likelihood of inappropriate behaviours or problems within project the team was at the very early stage of the project when they were assembling and building the project team. They intentionally picked people that were strong in the required values (agile values and the Ocean Group/River values). This practice helped them to have some prediction of the project teams' behaviours and prevent problems and conflicts as much as possible.

Amount of monitoring

Transparency is a major influence in reducing monitoring in distributed projects, which is why the Lake Project team used different tools and practices to increase the visibility and transparency among the project teams across locations. The required monitoring process is low in the Lake Project. BA from team S refers to this as *"Well the test for required monitoring would be if I disappear for a month, would the project fail? It wouldn't and it would keep going."* The required monitoring is very simple. For example if they want to check if there are any risks around delivery it is only through having a couple of conversations. The amount of monitoring is reduced in the Lake Project through practices like regular meetings and automated tools to measure code quality. There are regular meetings every fortnight at a program level to talk about any problems. There are informal chats and regular meetings to do monitoring. Some issues require meetings more often than others. For example, monitoring delivery is on daily basis but monitoring the strategy is over time.

Practices such as Stand ups are vital because team members rely on other team members to talk about what they are doing or raise any issues throughout the day as well. Stand ups are where it all begins, the purpose of a Stand up is to voice any issues and also talk about what they are going to be working on for that day. There is no need to have a high monitoring level in the project. But there are other tools to keep the visibility throughout the day such as JIRA as explained below.

Because of the distance between the team members of the Lake Project, the visibility was relatively lost at the beginning of the project. For example, a team member picks up a task and, after few hours, understands that someone else already working on the same task. So they decided to put safeguards around it to make sure that it does not happen again. The project team members used a virtual tool called JIRA to increase the transparency of project team members' activities across geographic boundaries and reduce the amount of required monitoring in the Lake Project team. JIRA is an agile virtual wall with a virtual view of all the tasks and it is accessible by all project stakeholders across locations. They can get into it and find out which task is assigned to which team member. However, at the beginning of the project they did not keep JIRA updated and as a result the task assignments were not updated and this caused confusion.

The challenging thing with JIRA is that you need to keep it updated all the time. It is especially more critical for team M because they rely on this wall very much compared to team C and team S – so much so that they have a physical wall besides JIRA. The reason for this, as explained by the project manager, is that team M members are the mainframe developers compare to team C and team S that are front end developers, and this makes it more essential for team M to make sure they are updated. As soon as a task is picked up by a team member, he has the responsibility to update the JIRA. They also introduced other safeguards, such as when a team member picks up a task they have to go and speak to the person that had the task last and read the acceptance criteria. This is called a sanity check. It means there is no need to keep monitoring every single story card if this has been done correctly every time.

The teams are encouraged to be transparent in their activities. They are always reminded to speak out and to bring up the issues in catch ups sessions, meetings and conversations. They even have Retrospective sessions as a specific tool for the project teams to bring out all the issues with the project. For example, the Lake Project IM remembers when some team members across locations approached him to openly talk about their issues such as not feeling they are being utilised to their full potential by being on that piece of work or this is not the piece of work they want to work on for these reasons. He explains "*The way we encourage this transparency is that when they bring it up and we always appreciate them for bringing that up and [let him know that] it's important that you bring that up as much as you can [and] let us know these things because we need to work through them.*" This example not only indicates the transparency but also indicates how the project team members feel safe to openly talk about their issues with the project IM.

Equity among members

The equity among the team members can be seen in the decision making process – everybody has a voice in decision making. The decisions are not made by the individual project team members, even senior roles like the BA lead and tech lead are not the individual team members who make the decision but they bring the decisions together. The Lake Project IM refers to this as *"There is no king or queen in the decision making process."*

Agile values are another indicator for having equity and autonomy among the team members, IM, business analysts (BAs), testers (QAs) and developers, all sitting at the same level in the team across locations. *"It's more about agile values that hierarchy is not the concern but getting the work done."* (BA, team C)

The Lake Project teams believe that they are treated equally and they treat each other equally, "*We all treat each other really equal*." (IM, Lake Project). There were times that something was not fair, but they always tried seeking balance. If a project team member gets an interesting task, they make sure that the other team members get the opportunity as well. *"If team members get the opportunity of fairness, there is a balance. There are cool things inside the project team and things that are not so cool or team members don't have a passion for but we always try to keep that fair."* (IM, Lake Project)

It is a very delicate and sensitive job to provide equity among the team members for a large distributed agile project such as the Lake Project. Although the Lake Project team members feel equal to the other team members, there are instances that have made the equity quite challenging in the team. For example, there was a case where the project team members felt that they should be all part of Showcase meetings but the whole team was not invited. The team members felt to some extent this was unfair, but it was around understanding why they are doing it this way. The reason behind it is the Showcase meetings are not the agile Showcase meetings in a true sense; it's more about managing expectations of the business and the program. A developer explains *"That was the gap in knowledge and understanding of what the showcase was about. So once it was explained what the purpose of the Showcase meeting was, we moved on."* (Developer, team S)

Trust-based interactions

There is high level of trust among core team across geographic boundaries as well as between the core team and the business. The Lake Project teams were aware of the fact that it was very important to build that trust with stakeholders and with the people (in this case e-commerce people) who are going to pay for this project. The Lake Project manager believes *"There is high level of trust between the core and the business because of the fact that they are still willing to pay for the project and they trust the Lake Project team members can deliver."*

There are two ways that the teams show they are trustworthy; by making sure they do not betray the team or distribute negative gossip and by fulfilling their promise to the other team members. But one of other the elements that indicates a high level of trust within the core team in all locations was the fact that everyone could very openly and easily talk about their issues and ask for help if they need. *"Based on people's openness you can tell to what degree members trust each other."* (BA, team C)

A simple example of a task that has been done based on trust would be when a business analyst elaborates a story in Sydney and gives it to a developer in China knowing that it will get delivered. The Lake Project IM refers to another major example where a big piece of a job was done based on trust across geographic boundaries and without trust between the project team members, it was impossible to achieve the final integration. He explains *"One of those big pieces of work was supposed to be integrated through to the backend, we had six months to go and we've done 30% of the project. It was in a situation that team C in China did not have any visibility of that. This is usually not the way we work and we are usually on the same page but in this particular case the China guys could not be on the same page for a number of reasons and they just trusted us and followed us and we got it in the end."*

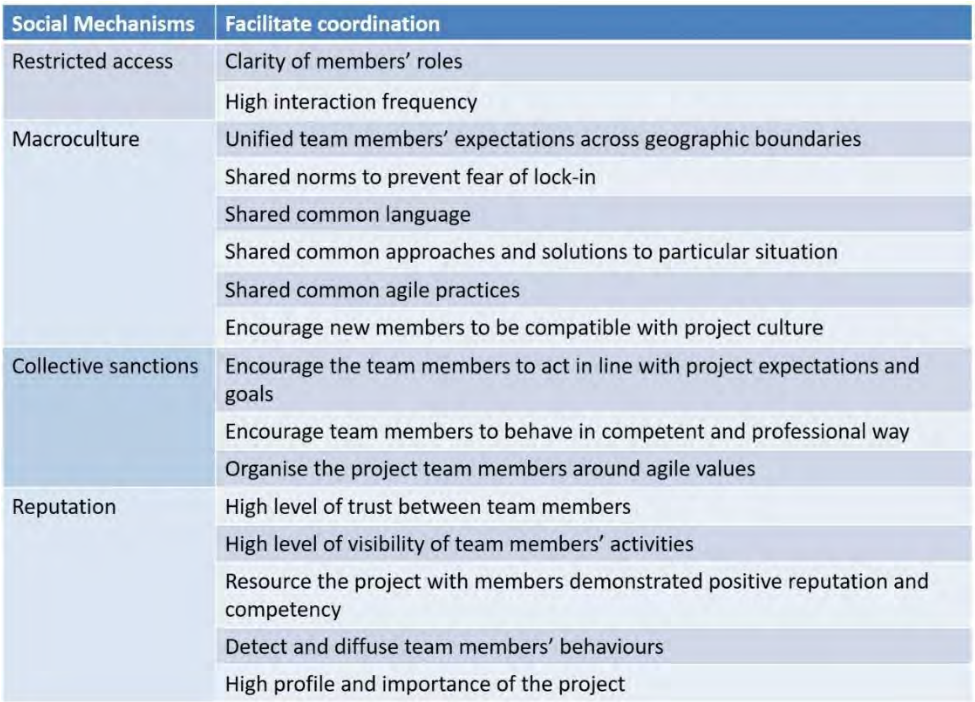
This example indicates that there is a high level of trust among the project the teams across geographic boundaries and that is why they are able to complete the integration transaction job. Such a trusted environment creates a 'safe-to-fail' environment for the project teams. The reason is that when the team members across locations feel safe to voice any problems or issues, there is no fear of lock in anymore and they start sharing their issues. Having a trusted environment is especially critical for a large distributed team such as the Lake Project because distance itself is a promoter for creating a non-'safe-to-fail' environment unless there is strong bond and high level of trust among the team members.

For example, there was a misunderstanding between team S and team C and a piece of software that was developed in China was not up to where team S thought it was going to be. A developer explains *"The reaction was like 'that's not exactly what want it, it's broken it's very much ok, we don't care who broke it, let's just find out why it broke so it will not happen again and let's try to fix it.' It's a fail safe environment."* (Developer, team S) By repeating such behaviours during the project life, the project teams get to know that this is the environment they are working in and they feel more encouraged to speak out when things are going wrong. An existence of a safe and trusted working environment prevents the blame game and protects transaction among the project teams across geographic boundaries.

5.3 Research Question 2

The second research question is 'How do social governance mechanisms coordinate exchanges among distributed agile development project teams?' The question investigated how social governance mechanisms facilitated the coordination of exchanges in a distributed agile development projects. To address this question this study conducted an in-depth investigation into social governance mechanisms' impact on coordinating exchanges among the Lake Project teams. As result of this analysis, I found a strong relationship between restricted access and coordination, macroculture and coordination and finally between reputation and coordination. However, the analysis presented a weak relationship between collective sanctions and coordination. The following table summarises the findings, then I explain these relationships in more detail:

Table 5. 2 How do Social Governance Mechanisms Facilitate Coordinating Exchanges in Distributed Agile Development Projects?



Social Mechanisms	Facilitate coordination
Restricted access	Clarity of members' roles
	High interaction frequency
Macroculture	Unified team members' expectations across geographic boundaries
	Shared norms to prevent fear of lock-in
	Shared common language
	Shared common approaches and solutions to particular situation
	Shared common agile practices
	Encourage new members to be compatible with project culture
Collective sanctions	Encourage the team members to act in line with project expectations and goals
	Encourage team members to behave in competent and professional way
	Organise the project team members around agile values
Reputation	High level of trust between team members
	High level of visibility of team members' activities
	Resource the project with members demonstrated positive reputation and competency
	Detect and diffuse team members' behaviours
	High profile and importance of the project

Restricted access's impact on coordination

I found that the existing level of restricted access in the Lake Project enhanced the coordination among the teams across geographic boundaries through the high level of interaction frequency and by having clearly defined roles including coordinator roles in the project. The following paragraphs provide more detail.

In the Lake Project the number of business and IT people (core team) that could potentially interact with each other is relatively large. However, by significantly restricting the number of project team members that collaborate to produce the software, coordination is eased and the uncertainty of future exchanges is reduced. I found that restricted access facilitates coordination of exchanges among the Lake Project teams across locations through high interaction frequency, high levels of visibility and a high clarity of roles. The existing restricted access in the Lake Project meant fewer project team members interacted for certain types of exchanges and this led to a high frequency of interactions among those certain team members for that specific type of exchange. This continued exchange among the Lake Project teams across locations motivated team members to learn about each other's way of working, share their knowledge, build trusting relationships and develop better mutual interests, which are very critical for a distributed project such as the Lake Project. Thus, it increased the motivation and ability of the project teams to coordinate smoothly across geographic boundaries.

I also found that there is high clarity of roles in the Lake Project team across locations. The Lake Project IM says *“Each role in the project is clearly defined and everyone is very well aware of each role’s responsibilities and accountability”.* Such high clarity of roles in the project makes it easier for the project team members to identify with whom they need to interact if they have an enquiry. This enhances coordination among the Lake Project teama across geographic boundaries because team members’ interactions are more focussed, faster and more efficient. However, although the roles are clearly defined, you still find a lot of fluid roles exist in the core team and this is because of the agile nature of the project. The core team members work as cross functional teams across locations and they sometimes overlap with each other.

In addition to a high clarity of roles, I found that because the Lake Project team has large number of team members distributed across different locations, the project allocated additional points of contact at each level such as the program level, project level or the core team level. These points of contact (lead BA, lead tester or tech lead) are responsible for coordination among certain Lake Project team members across geographic boundaries. Restricting interactions through these points of contact across locations made communication easier and enhanced coordination by having less work interruption and less work and communication overload. The following sections set these findings out in more detail.

High Interaction Frequency

As discussed in application of restricted access, things like status maximisation, ideological similarities, relational contracting and embeddedness among the Lake Project team members facilitate more focussed interactions among fewer team members which leads to a high level of contact and exchange among certain team members base on their status, interests and the strong ties among them. This saved time, effort and costs to communicate, especially for exchanges across geographic boundaries, therefore it enhanced coordination among the Lake Project teama across locations. *“There is absolutely a high degree of interaction among team members, constant talk. In the Lake Project, as it is distributed in four locations, we’ve got the business in Brisbane, so from a delivery team perspective the interaction is a lot”* (Project manager, Lake Project)

The Lake Project team members prefer to interact frequently with other similar status team members to avoid confusion, save time and be more focussed. The team members would like to exchange with the other team members across geographic boundaries who have a similar status or level. For example, during the Showcase meetings only the higher status team members of the project, like IM, the team leader, program sponsor and portfolio manager, participate and interact, rather than the core team members. "A*lthough sometimes there are some representatives from the developers also, it is very limited. Higher status members of the project do not want to have a lot of interruptions and technical details in Showcases."* (IM, Lake Project) Thus, the technical project team members' participation is very limited. Indeed, this was the main reason the technical team members of the project started the Homemade Jam meetings for themselves only. They would like to interact with other technical team members that have a similar status and interests. This example indicates that by limiting exchange to among similar status team members, the Lake Project team was able to enhance the coordination among the teams with similar status across locations.

One the other hand, BAs prefer to have more frequent interactions with each other otherwise coordination would be challenging. The team C business analyst explains *"Having high level of contact between BAs is required otherwise as a distributed and large team like we have, the BAs spend a lot of time on fragmented tasks."* If they didn't have a really strong channel between the BAs, everything would become fragmented and no one would have the visibility of the other team members' status.

In addition, restricting interactions among certain statuses such as lead tester, lead developer and lead BAs to their dedicated forums facilitates a high frequency of interaction among certain team members more than others. The same applies to BA Forums, Developer Forums or Tester Forums. In each of these forums only the team members with similar roles are interacting. For example, in Developer Forums, there are developers only who interact frequently across geographic boundaries and discuss issues. Such high frequency of interactions with fewer team members who have similar interests makes coordination much easier for all the teams across locations.

Due to the strong ties and embeddedness among the Lake Project teams everyone knows each other and everyone knows about each other. The strong bond between the project teams across locations motivates them to have more frequent interactions. This makes the coordination among those teams much easier.

Furthermore, the high level of interaction among certain project team members developed a common language, established routines and established a similar way of working across geographic boundaries. They set out routines and the way they want to run their daily Stand ups, Retrospectives and IPMs (Iteration Planning Meetings). This made coordination among the project teams across locations smoother and easier. For example, the Lake Project team leader explains *"Through frequent contact with China we understood they need extra sessions on context. Then we start developing weekly meetings between BAs and the China team to give them a bit of extra context especially if they missed out on important workshop."* (Team leader, Lake Project) The extra context sessions for team C become a routine through the project life.

There was very rich interaction between team S and team C on daily basis. This was due to the fact that team C and team S were both front end developers and the nature of work was much more similar than team M, who were backend mainframe developers. These rich and frequent interactions among team C and team S facilitated the development of common languages and approaches. For example, they use words commonly used and familiar between the teams. For example 'IPM' being an abbreviation of Iteration Planning Meeting and 'Sync' being an abbreviation of Synchronisation.

The above indicators of restricted access mean fewer team members interact more often for certain types of exchanges which improves both parties' ability to coordinate smoothly. When fewer project team members are involved in exchanges, they interact more frequently. These constant and frequent interactions allowed the Lake Project team members to get to know each other's expectations and get familiar with other team members' skills, goals and their abilities across locations which enable them to coordinate easier. In addition, through these frequent interactions, the Lake Project teams learned about each other's abilities and working environment and developed communication protocols and working routines across geographic boundaries.

Clarity of team members' roles

Coordination is achieved by organising the project structure around clearly defined roles at different levels responsible for certain types of exchanges. For example, at a program level there is the program IM, program manager, program sponsors and the program test lead. At the project level there is the project IM, project manager and the team leader. At the core team level there are testers, BAs, developers and the IM.

Due to existence of a large number of project team members in the project and distribution of the teams across locations, the project involved additional points of contacts at each level required to act as a coordinator among the project teams. For example, the project manager acts as a coordinator between project IM and program manager, or the program IM role is defined clearly to act as the coordinator between different project IMs and program managers, or the project IM acts as a coordinator between the team leader and the core team members. A business analyst explains *"Without having these roles, you would have a number of team members pulled in one direction because they've got to maintain a relationship with everybody and coordinate their issues. Those point of contact channels do stop one person just being pulled into million different directions by 27 people in the team."* (BA, team S) The existence of such coordinator roles enhanced the coordination among project teams.

For example, the core team members go to BAs to clarify requirements because BAs act as a mediator to talk to the product owner or e-commerce people. The Lake Project IM explains *"In a lot of scenarios when it's getting to the developer or tester they go to the BA because they recognise the business analyst can help us and understand the strong side of the business. So they say 'let's speak with the BA about the issue' so they'll go back to the BAs and say 'hey guys can you just go and speak to the business about this'."*

Coordinator roles that exist at a core team level include lead testers, lead BAs and tech lead as coordinator. They provide coordination between the core team members and the project IM. For example, instead of having a couple of testers or BAs interact with IM at the same time about some issues, the project core team members interact with lead testers or lead BAs who are a point of frequent interactions with the IM for them. Then they communicate the core team members' issues and concerns with the IM and get back to the core team members. It means less interruption to work and saves time and effort. This example indicates having additional points of contact as coordinators in the project team structure ensures the communication and coordination is less confusing and more focussed by reducing the communication overload.

In other words, such role clarification reduces the work load for each project team member and would lessen the interruptions at work which enhances the coordination among the teams. For example in the case of portfolio manager role, there is no need for the team leader to deliver the project for the portfolio manager. He has already got the project managers and program IM roles that coordinate to deliver the project. There are so many projects for him to keep an eye on. That's why he has got the team leader role that actually keeps an eye on those projects for portfolio manager. The Lake Project manager explains *"The portfolio manager needs very little contact and as a matter of fact if they get every possible contact from everyone, they got overloaded. So not every person needs to know so much context and detail and that is why we structure the project with so many roles so they can get responsibility for different parts of the project."*

An interesting example is when, during the Lake Project life, the original IM had to go on leave and the BA from team S replaced him and acted as the IM while the original IM was away from the project. He then returned to his original role as BA after the original IM returned to the project. He explained how his role as IM was clearly defined as different from his previous role as BA. His role as IM involved more frequent contact with less project team members (more focussed) and one level more removed from the details. *"I am not immediately across the fine details of each story and what's going on in that particular iteration, beyond what's blocked, who needs help, what are we actually going to be doing next week and what I need to do now to make sure that we can deliver the story next week. So my communication now is a lot more with the lead tester, the lead developer rather than individual people in the team."* (BA, team C) This example indicates how clearly each role's responsibilities and accountabilities are defined from other roles in the Lake Project.

In general, the project team members prefer to go through the structure channels and interact more frequently with certain other team members for specific type of interactions. *"If we need something we know to whom we should go."* (BA, team C) A developer from team S explains that *"Because we have clear defined roles of course we go for specific questions or interactions to the BA for example and if it's a question of testing I go to the testers and they do the same as well."*

The core team members recognise BAs as a better communication channel and refer to them if they need clarification. *"If they don't understand the requirements, they often go to the business analyst and say 'hey what this requirement about again?'"* (IM Lake Project) This example indicates that having certain team members identified for certain types of exchange enhances clarity and identification of their roles.

In addition, as explained before, at program, project and core team level roles are clearly defined and everyone knows what those roles are accountable for based on their core skill set, whether it's a developer, tester, BA or an IM. However, when it gets to the core team level, due to its agile nature and cross functional team interaction between team S, team C and team M, there are a lot of fluid roles as well. The team members move around and overlap with each other depending on what the day to day team problem is. For example they can just turn up in the Stand up and find out what's going on that day in 15 minutes. *"Everyone overlaps pretty well; everyone knows mostly what people are working on."* (Developer, team S)

BA from team C explains that *"If we notice that we've got a back log in testing and we need to help in some way because the testers cannot keep up with all the testings, to get to the outcome of the iteration, we usually put BAs and in some cases developers on the testing and clear that back log."* In an ideal agile team all the team members are cross skilled and can have cross roles but it's not that way in the Lake Project team. They only step out into another role if they need to help out and then they default back to their regular roles.

There is increased clarity of roles, responsibilities and accountability of each of the project team members across locations**.** The team leader believes the existence of these clear roles in the project structure saves time, effort and cost and makes the coordination much easier. *"Having certain point of contacts in different parts of the project makes it easier to coordinate and saves a lot of time and effort to reach them. So these people are points of contact to make it easier, otherwise they will be overloaded with lots of interruptions."* (Team leader, Lake Project)

The Lake Project IM adds that the project structure made it much easier to resolve issues in this distributed project. *"The way we structured the project team with all these clarified roles makes it much easier to reach to others, especially across locations. We've got to get resolution on a lot of things, knowing what responsibilities sits before the person. So, it's a lot quicker and easier to get resolution on any issues if I go to the right person."* (IM, Lake Project)

Therefore clearly defined roles restrict access to certain project team members for certain types of exchange depending on the exchange enquiry. The clarity of these roles made it much easier for the project team members to identify which team members they need to access across locations for certain enquiries which made coordination smoother for them.

Macroculture's impact on coordination

I found the existing strong macroculture activated through different practices enhanced coordination among the Lake Project teams across locations very well. The following paragraphs provide more detail.

In general, shared common norms, values, goals and assumptions encouraged a similar way of working across geographic boundaries. Similar ways of working across locations facilitates coordination among the teams across locations. This is achieved through developing a lot of common functionality, code reviews, tacit rules, common language and common approaches that are shared between these Lake Project teams across geographic boundaries.

I found that the Lake Project macroculture enhances coordination among the project teams in six different ways: (1) by unifying team members' expectations across geographic boundaries; (2) by sharing common language among project team members across geographic boundaries; (3) by sharing common approaches to particular situations; (4) by sharing common agile practices such as pair programming and daily Stand ups; (5) by sharing strategies to prevent fear of lock-in (i.e. voicing out, no blame games); and finally (6) Encouraging new team members to be compatible with project's culture. The following sections set these findings out in more detail.

1. Unifying team members' expectations across geographic boundaries

Unifying team members' expectations was achieved through socialisation, context sharing, shared goals, shared norms and increased visibility of activities so the team members do not work at cross-purposes on different sites. The project team members socialise across geographic boundaries through daily Stand ups, Retrospectives, Showcase meetings and the forums. In every one of these socialisation sessions, the team members are able to raise issues, ask questions to resolve the ambiguity and be updated about the other team members' ideas and activities. "*They are able to become familiarised more and more with other teams and their way of working, learning more about context and becoming more aware of expectations and goals."* (IM, Lake Project) These socialisation practices encourage similar ways of working across locations and enhanced coordination among the teams.

One of the practices that the Lake Project used to harmonise team C with the other two Ocean Group teams (team M and team S) was extensive workshops, classes and extra activities in addition to their usual work. These activities were English classes, technical courses, learning about Australian working culture through socialisation with the other two teams and having extra context sessions. *"These activities provided similar way of working on different sites and harmonised the three teams' way of working, goals, values and expectations across locations."* (Team leader, Lake Project) This improved and eased the coordination among the teams.

Another effective practice to share the context was to co-locate teams from different sites at the beginning of the project and frequently throughout the project life. Team S, team C and team M were co-located in Sydney or China and went through extensive context workshops and socialisation sessions. "*This was a great opportunity for all team members to get the same level of understanding of context."* (Agile coach, Lake Project)

The context of the shared concepts among the teams was not only the technology concept but it rather covered a wider concept. *"The shared context is not just the context around the platform itself but the context around the structure of the company and getting to know how everything works."* (Team leader, Lake Project) Therefore, providing that context, especially for team C, proved to be very valuable because they start having the same type of processes and driving towards the same goal. This practice reduced the likelihood of the team members working at cross purposes across locations. Having a similar understanding of the context among the Lake Project team members across geographic boundaries smoothed the interactions and enhanced coordination between them.

Shared common goals across locations lessens the possibility of working at cross-purposes. As discussed previously, there are four major goals that are shared among all the project teams across geographic boundaries. These goals are delivering the project on time, making the distributed agile component work, delivering the project technical strategy and delivering software to the satisfaction of the business.

These shared goals among all the project teams provide a closer relationship and cultural alignment between the teams in Sydney, Melbourne and China. It is an assurance that they are all working towards the same goals and they are on the right track. It provides a mutual adjustment among the project teams that encourages convergence of expectations among the teams across geographic boundaries. For example, there are certain expectations from the Ocean Group and the River partnership which are very well communicated among the teams through shared goals. One of the reasons that the Ocean Group made the partnership with the River for this high profile project is the embedded culture of agility in team C. A developer explains "*The Ocean Group teams and team C are sharing this goal that they are expected to make this distributed agile project works in all three locations."* (Developer, team S). This facilitates a harmony in the working environment that means a lesser likelihood of cross-purpose working, thus easing the coordination among the project teams across locations.

For example, a business analyst explains that if a project team member doesn't understand and share the goals, it would be difficult for them to determine what they are supposed to be doing and it increases the possibility of "rework". He adds "*Shared project goals literally made us have no rework in our team. Also plenty of our work is pretty aligned because we've been working towards the same goal and we've been working together rather than separately."* (BA, team S) They used the tools (i.e. JIRA) that are available to them to make a shorter path because *"We always know what the other side is doing, where they are. So if you look at the screen we know at the wall what is happening."* (Developer, team S). This example also indicates how shared goals among the teams across geographic boundaries provided a similar way of working on different sites and protected the team from working at cross-purposes.

One of the most important practices that is used in the Lake Project to align expectations and activities of the project teams is to increase visibility. They use JIRA so every single project team member has the visibility on project progress across geographic boundaries. Such visibility makes it much easier for the team members at different sites to work on the same page, not at cross-purposes, thus enhancing the coordination among the teams across locations.

2. Sharing common language

Shared common language (stereotype language) and terminologies developed at the beginning of the project among teams, allows easier coordination. *"Because we know and share a context and we form a language around that context."* (IM, Lake Project) Beside this, there are some common language and terminologies developed throughout the project life to simplify the communication of complex data and information.

For example, there is a feature where the Lake Project team members want to be able to send data from their application to an external application to submit the application details. The external company doesn't provide another means to allow them to easily transfer data. So when they are looking at a solution to pass that information on, a business analyst explains that *"We will pass it to the user interface of that external application, now that solution, it's a bit of a tricky one and we kind of simplify that to a terminology called Screen Squirting. So it's like getting the data from one application, squirting it to another."* (BA, team S) They are using this terminology rather than explaining it over and over again. *"We rounded it down to this simplified term."* (Developer, team S)

Another example is a feature around catching a customer's address details. Addresses can be written in various formats that is why capturing the user's address is one those pieces that can be difficult to do especially when you are trying to capture the address very accurately and in a certain format. The Lake Project team members need to make sure they capture the address details in a particular format and they have identified multiple paths and scenarios in which that data can be entered. "*Rather than going through different paths and explaining what those paths are we basically write them down and we call number 1 as the happy path and that's the most straight forward where the user entered it all right, then we have got sad path where the users entered it in completely incorrectly and then we should right the validation messages."* (BA, team C) They developed different terminologies to identify levels of address details that can be captured. For example, the team used the terminology such as 'sad' path, 'happy sad' path and 'sad sad' path. Terms like that helped to explain and transfer information much faster rather than explain it over and over again.

Other ways of simplifying complex information in the Lake Project is using pictures. Pictures like a process flow image helped them quite a lot to share and transfer complicated information across. For example, a business analyst explains that *"I'm working on a task now which is quite complicated and needs to be shared with another member who is away for two weeks. So I draw a process flow and that helped because it simplified it. I asked if she wants to catch up for an explanation but she said 'no it is OK I understand that, it is better than reading 15 pages of words'."* (BA, team S)

The above examples indicates developing familiar common language and terminologies to simplify complex terminologies makes communications among the teams smoother, faster and simpler. Thus enhancing the coordination across geographic boundaries.

3. Sharing common approaches to a particular situation

Throughout the project, the teams developed and shared tacitly understood rules to act appropriately in specific situations. For example, if the teams find out there is a lot of work left for delivery *"Everyone knows what happens. You go to planning and you get the business into the room and working out our approach to it."* (IM, Lake Project)

All the teams across locations are very well aware and know if a piece of work doesn't pass the test, the tacit rule is that the tester meets with developer to explain why it did not pass the test instead of throwing it back to developers and just asking them to correct the work.

Another example is measuring the number of days that the teams work on a task across locations to get an idea of how close teams are to completing tasks. The task could be 25%, 50% or 70% complete. The tacit rule is that *"If something is in 25% you can possibility have another developer jump on it and pick up something new, it's 25% and it is quiet small. It drives a lot different conversations and improves the coordination."* (Developer, team S)

A further example is if the project teams notice that they have too many blockers on the agile wall, it is a broadly understood tacit rule that *"Someone needs to say something and usually there is someone to raise the hand and say hey guys there are these blockers on the wall already."* (BA, team C)

The above examples indicate that the common approaches to certain problems made it easier to deal with the issue and eased the communication and coordination among the Lake Project teams.

4. Sharing common agile practices (like pair programming and daily Stand ups)

This provides a similar way of working across all locations among the project teams and makes coordination among the teams easier. The Lake Project teams share similar ways of working on different sites (agile), which eases the coordination. Agile provides tools to support the teams to work consistently across all locations. Indeed, having shared agile practices aligns the teams' activities at different sites. *"There are words that come from agile and can provide that consistency of language, protocols, how we run Stand up, there is a protocol about who speaks, how we run, what we talk about, elaboration sessions as well."* (IM, Lake Project)

The above discussion suggests that macroculture reduces coordination costs by increasing the ease of communication and coordination among the project teams across geographic boundaries. However, the Lake Project teams found a slight misalignment due to different cultural workings between China and Australia. Team C tended to work longer hours than team M and team S in Australia, even on Fridays, which for Australians is a relaxed working environment getting close to weekend. Team S and team M collectively work till four pm on Friday afternoons and team C did not understand the concept of a slow Friday afternoon, so there was to some extent a clash around that. A business analyst explains "*China team end the work on Friday like any other week day which is about 6:30pm or a bit later. They found it very strange that a team will get to 3 or 4 o'clock on Friday*" (BA, team S)

5. Sharing norms to prevent fear of lock-in

As explained before the ease of speaking out and no blame games are repeated behaviours (norms) that are shared among the Lake Project teams across locations. These shared norms make it much easier for the teams to be able to speak out about problems and failures without having the fear of being blamed or being locked out. Therefore, the teams are motivated to communicate when it is required. Such an environment encourages smooth coordination among the teams across geographic boundaries.

6. Encouraging new members to be compatible with project culture

The Lake Project new team members are encouraged to be compatible with the project culture through shared norms such as the shadower and buddy role. As explained previously, shadower roles are backup for main roles in the project in case a team member is unable to come for different reasons like being sick, on leave or needs to be changed. The norm of having shadower roles in the project ensured the Lake Project did not need to add a brand new team member to the project without any cultural support, which itself prevented complications and challenges of coordination. Alternatively, the norm buddy role is used when there is a new team member to the project team who is required to fit into the project way of working (culture). The new team member starts working on a task while they have a senior team member (a buddy) supporting them. This makes coordination of new project team members' exchanges much faster with rest of the team across locations.

In addition, I found that the Lake Project teams are able to create a sense of mutual interest between the teams across geographic boundaries through socialisation activities such as co-locating distributed teams throughout the project life cycle. During several co-locating practices, the project teams shared their culture, skills and experiences, which opened the door for the teams to have a better understanding of each other's interests. Thus, such understanding and awareness eased the coordination among the teams across borders.

Collective sanction's impact on coordination

I found that collective sanctions in the Lake Project facilitated coordinating the project teams' interactions across geographic boundaries. The collective sanctions coordinate exchanges in the Lake Project by (1) encouraging the team members to act in line with project expectations and goals through the high level of transparency and visibility of the team members' activities; (2) encourages them to behave in a competent and professional way through shared norms including peer pressure, no blame game and by promoting and rewarding acceptable behaviours; and (3) organising the project team members around agile values.

1. Encouraging the team members to act in line with project expectations and goals

The Lake Project teams across geographic boundaries are motivated to act in line with the expected behaviour by increasing the visibility of what the project expectations are. They are given a vision of what the project wants to achieve and how the project wants them to be aligned with that vision, otherwise they have to leave the project. The project manager explains *"Visibility and transparency of activities is very critical in the Lake Project distributed environment to coordinate members' interactions. It helps avoid conflicts and confusion, it also saves time and effort among team members across all locations."*

In addition, the project members are encouraged to be transparent in their interactions with each other across locations using JIRA and voicing out practice. JIRA has the ability to make the teams' activities visible across borders. The Lake Project also encouraged project teams to communicate and voice out issues as soon as possible. These practices increase the visibility, motivate trust among the team members and encourage them to be transparent in their dealings with the other team members, thus easing the coordination of their exchange across geographic boundaries. Furthermore, the rewarding actions and incentives for appropriate behaviours, positive attitudes and high performance in the project have a positive impact on coordinating the project teams' exchanges across locations.

2. Encouraging team members to behave in a competent and professional way

I found that the Lake Project teams are encouraged to behave in a competent and professional way through rewarding acceptable behaviours and shared norms such as peer pressure, no blaming game and meta-norm in order to facilitate coordination across geographic boundaries.

Rewarding acceptable behaviour: the Lake Project promotes positive behaviours and recognises acceptable behaviours among the team members, so the team members are tempted to adopt and engage with those behaviours within the team more often. The reason for their motivation is that the acceptable behaviour or high performance is regarded as a very positive activities by the teams across locations and gains their trust and respect. This practice has a positive impact on coordinating the project teams across locations. A developer explains that *“When you see another member is rewarded and regarded highly because of professional behaviour, you will try your best to behave similarly to gain the trust of the rest of the other members as well.”* (Developer, team S)

Peer pressure: This informal monitoring practice, locally and across geographic boundaries, increases the potential of positive behaviours and brings more discipline within the team across borders. The project team members are watching or following other team members’ activities on daily basis as part of their daily activities in order to be updated. Meanwhile if they notice something is going wrong they voice it out. In this regard, a business analyst gives an example, *“For example, I noticed somebody struggling a bit, I go, ‘is everything all right? What can I do?’ I ask out of concern. So we have peer pressure to raise underperformance and peer questioning of the individual.”* (BA, team S)

This informal monitoring is often in practice during pair programming across geographic boundaries, where a team member is constantly monitoring the other team members’ work. If they notice inappropriate behaviour they raise it as a concern. The same applies to shoulder check and code review practices. A business analyst argues that *“In this way we manage to have a much more disciplined team and behaviours are more aligned with the Lake Project expectations.”* (BA, team C)

No blame game: the project team members jointly deny or prevent blaming game and negativity across geographic boundaries, which encourages the team members to behave professionally. When the issues are raised early without fear of being blamed, then it can be taken care of much easier and faster before it causes complications in coordination among the teams.

Meta-norm: meta-norm encourages the team members, in particular senior roles, to constantly monitor and act against unacceptable behaviours of the other team members, otherwise they will face the consequences. In this regards, the Lake Project IM explains that if they do not act accountable for actions and activities of the team members that they are responsible for *“We are questioned why we did not take care of the situation, why we did not monitor the case and did not report it on time”.* Similarly, the Lake Project team leader adds *“As a lead you should be looking at monitoring inappropriate behaviours and you should be looking at having a more positive attitude.”*

3. Organising the project team members around agile values

The Lake Project organises the project team members around the project values including agile values. Particularly when unacceptable behaviour occurs, the teams are reminded of the values. Otherwise the project team members have to go through gradual escalation. This encourages a similar way of working across locations, thus enhancing coordination among the project teams.

The above norms and practices encourage the teams to act competent and professional across all locations, thus easing the coordination of their exchanges. I therefore conclude that collective sanctions, as well as restricted access and macroculture facilitate coordination of exchanges.

Reputation impact's on coordination

One of the key findings of the case study is that reputation facilitates coordination of exchanges. Indeed, the results of the case study suggest that all four social mechanisms – restricted access, macroculture, collective sanctions and reputation – enhance coordination of exchanges among project teams across locations. Reputation facilitates the coordination of exchanges across geographic boundaries through the existence of a high level of trust between the project team members, resourcing the project with team members that have demonstrated positive reputation and competency, detecting and diffusing the team members' behaviours, high visibility of the teams' activities and the high profile and importance of the Lake Project. These practices encouraged the project team members to maintain their positive reputation, thus having more tendency to coordinate. As discussed before, the project team members across borders perceived social and individual expectations as an important part of their character. This motivates them to demonstrate and maintain a positive reputation. Therefore, such tendency encouraged them to coordinate much easier with the other team members, enhancing coordination across geographic boundaries.

1. High level of trust between members across locations

I found that the project teams' interactions across geographic boundaries are enhanced due to the existence of trust in other team members' ability to perform tasks based on previous presentations of their quality of work, i.e., their reputation. Building and maintaining a positive reputation along the way with the other project team members allowed others to trust their ability to perform the tasks more and this enhanced coordination among them.

For example, one of the reputation based interactions across the team was when team C and the Australian teams had to integrate a major part of the project through the backend and they were not able to provide any visibility to team C. However, there was a lot of trust across the teams because of the previous quality the project team members had demonstrated (positive reputation), which allowed team C to follow the Australian teams to develop this major part of the project. The project team members paired in China and Australia to do the task that would have been impossible without the trust based reputation shared among the team members in such a distributed project. This example indicates trust based reputation between the team members enhances coordination of exchanges among the project teams across locations.

2. Resourcing the project with members who have demonstrated positive reputation and competency

I found that the practice of resourcing the project with competent and professional individuals based on their particular skill set and reputation made the teams expect to move things along quicker because they already had the context and the knowledge.

In general, the Lake Project team members are concerned about the skills and experience of the other team members when they want to interact with them and in particular, when it comes to asking to complete a task. Then trustworthiness comes into the picture and the team members would like to know if the other team member have a reputation for delivering their tasks. The Lake Project IM says *“You do notice that there is a tendency for certain people to go to a particular person in the Lake Project for certain exchanges and enquiries. We know the background based on what the team leader advised us.”* Indeed, there was an expectation that the team members will fulfil their responsibilities based on their reputation and this helped the team to get along much easier. It saved them time and effort and improved the coordination among them.

3. Detecting and diffusing team members' behaviours

I found that the strong embeddedness that exists among the Lake Project members across locations enhances the flow of effective information about the team members' behaviour. Therefore, the project team members are well informed about the other team members' reputation, trustworthiness and capabilities. However, as mentioned before, the Lake Project encouraged spreading positivity in order to maintain the trust and the positive environment among the teams across borders. This also enhanced the existing ties and embeddedness among the teams, thus facilitating the coordination of exchanges among project members across locations. I found that negative reputation complicates coordination as the team members hesitate to interact with non-competent team members.

For example, the Lake Project team members are more eager to interact with the other team members that they know they have the skill set and knowledge they are looking for because this will save them time and effort and improve the coordination among them. For example, a developer indicated that *"Majority of time I prefer to interact with team members that I know [reputation wise] whatever I ask about that subject, it's probably going to be quicker, in particular when I am in a rush. I definitely think other people in the team do the same thing."* (Developer, team S)

The China team business analyst also believes it is the same case across geographic boundaries and in all three locations, including China. For example he explains that *"We have two developers who are very knowledgeable on the code base and some of the business logic and rules behind it. There is also one more developer that worked on that particular context a lot longer. If I need to respond to the business I would definitely take his knowledge into consideration and I am more inclined to go to the developer with longer experience on that code and context."*

The above examples indicate that when the Lake Project team members find a team member with a positive reputation on a certain subject, they prefer to refer to that team member about the subject and this helps the coordination among the teams in a positive way. It is a similar case if a particular project team member does not have a good reputation; the other team members across locations hesitate to interact with that person at the first place.

4. High visibility of team members' activities

I found that the existing high level of visibility and transparency of the project teams' activities to the other teams across geographic boundaries motivated them to act as competent as possible to gain or maintain their positive reputation. The Lake Project IM says *"When team members see their activities are visible to all project members and there is nothing to hide, they do their best to complete their tasks as good as possible to gain and maintain positive reputation because then other members can trust them more."* (IM, Lake Project) This example indicates that high visibility is an enabling factor for reputation to facilitate coordination. In addition, transparency of the team members' activities also encourages them to have their activities aligned with the norms, values and goals of the project. Thus, this harmony of activities across geographic boundaries enhanced coordination of their exchanges.

5. High profile and importance of the Lake Project

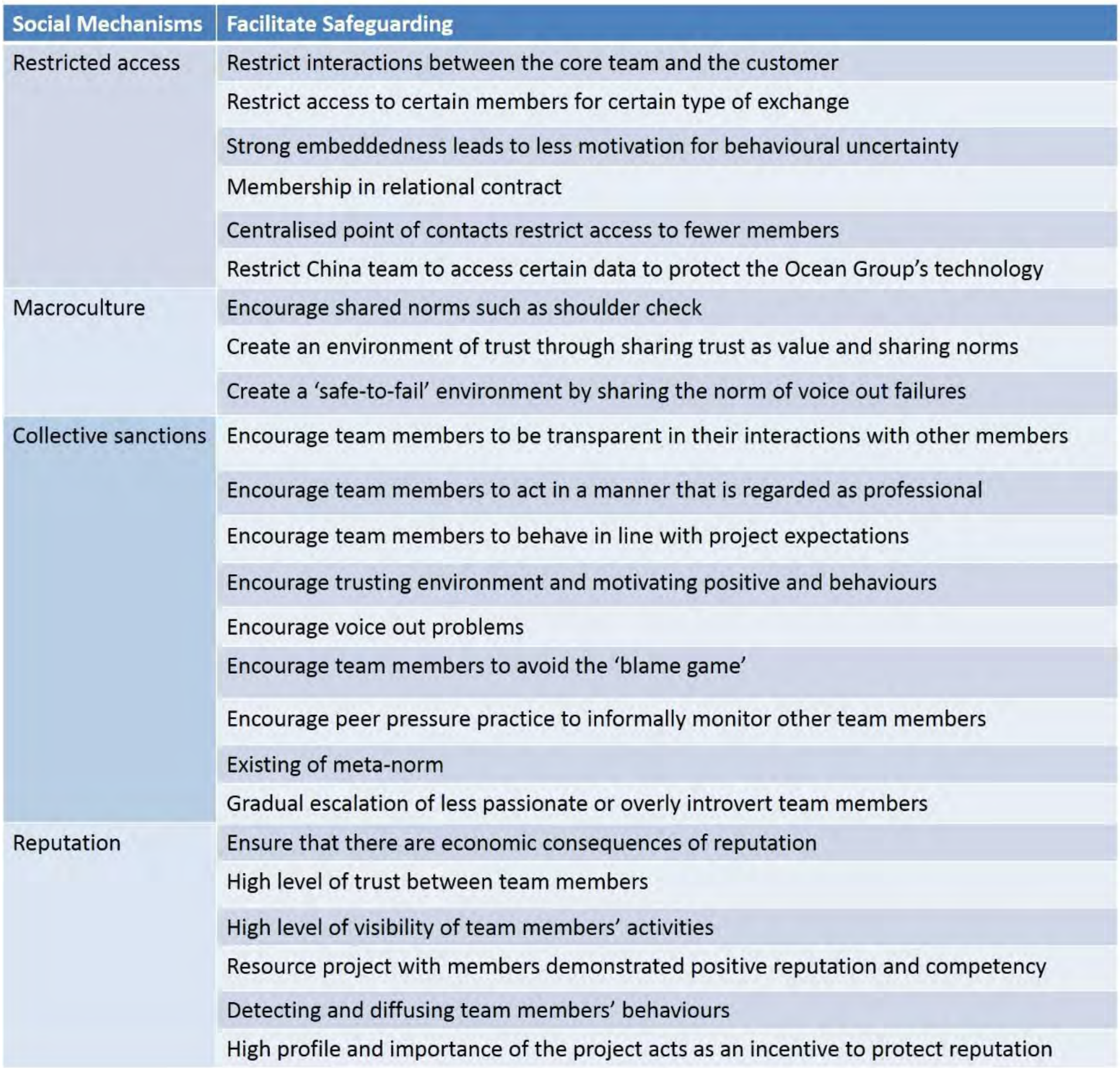
I found that one of the major reasons that the Lake Project team members across borders have a strong tendency to gain/maintain positive reputation is because the Lake Project is one of the most important and high profile projects within the Ocean Group and wider network. For this reason, the team members across geographic boundaries have a strong desire to stay a team member of the project in order to enhance their own career. A developer explains *"Having positive reputation in such a high profile project is considered very important by team members for their current and future career."* (Developer, team S) This motivated the teams to coordinate their interactions across locations.

I conclude that the above practices encourage the team members to gain/maintain positive reputation across all locations, thus easing the coordination of their exchanges. I therefore conclude that reputation, as well as other social mechanisms, facilitate coordinating exchanges.

5.4 Research Question 3

The third research question is 'How do social governance mechanisms safeguard exchanges among distributed agile development project teams?' The question investigated how do social governance mechanisms facilitate safeguarding exchanges in distributed agile development projects? To address this question this study conducted an in-depth investigation into social governance mechanisms' impact on safeguarding exchanges among the Lake Project teams. As result of this analysis, I found there is a strong relationship between restricted access and safeguarding, macroculture and safeguarding, collective sanctions and safeguarding and finally between reputation and safeguarding. The following table summarises the findings then the study explains these relationships in more detail:

Table 5. 3 How do Social Governance Mechanisms Facilitate Safeguarding Exchanges in Distributed Agile Development Projects?



Social Mechanisms	Facilitate Safeguarding
Restricted access	Restrict interactions between the core team and the customer
	Restrict access to certain members for certain type of exchange
	Strong embeddedness leads to less motivation for behavioural uncertainty
	Membership in relational contract
	Centralised point of contacts restrict access to fewer members
	Restrict China team to access certain data to protect the Ocean Group's technology
Macroculture	Encourage shared norms such as shoulder check
	Create an environment of trust through sharing trust as value and sharing norms
	Create a 'safe-to-fail' environment by sharing the norm of voice out failures
Collective sanctions	Encourage team members to be transparent in their interactions with other members
	Encourage team members to act in a manner that is regarded as professional
	Encourage team members to behave in line with project expectations
	Encourage trusting environment and motivating positive and behaviours
	Encourage voice out problems
	Encourage team members to avoid the 'blame game'
	Encourage peer pressure practice to informally monitor other team members
	Existing of meta-norm
	Gradual escalation of less passionate or overly introvert team members
Reputation	Ensure that there are economic consequences of reputation
	High level of trust between team members
	High level of visibility of team members' activities
	Resource project with members demonstrated positive reputation and competency
	Detecting and diffusing team members' behaviours
	High profile and importance of the project acts as an incentive to protect reputation

Restricted access' impact on safeguarding

I found the level of restricted access that exists in the Lake Project team facilitates safeguarding of the teams' exchanges across geographic boundaries through limiting the number of the team members involved in certain types of transaction. The following paragraphs provide more detail.

I found that in the Lake Project, restricted access facilitates safeguarding of exchanges across geographic boundaries. It does this by decreasing the number of the project team members involved in the exchange through a centralised point of contact, restricted access to certain team members for certain types of exchanges, strong embeddedness among the Lake Project teams and mutual adjustments through co-locating project teams. The interaction among fewer project teams across locations reduces the amount of monitoring required for exchanges. This lessens the behavioural uncertainty and the likelihood of opportunistic behaviour across locations.

I also found that having fewer team members involved in exchanges leads to more frequent interactions among the teams across geographic boundaries. This allows trust and strong relationship to develop among those teams across geographic boundaries. When this happens, the project teams involved in the exchange see their interests aligned rather than in opposition. Therefore, they have less motivation for opportunistic behaviours. In addition, I found when the Lake Project teams have continued interactions across geographic boundaries and they expect to continue these interactions in future, their level of trust and courtesy increases and their incentive for behavioural uncertainty and opportunisms in their exchanges reduces. Furthermore, I found restricted access also protects the Lake Project interactions through their membership in the relational contract. This membership allows the Lake Project teams to work in line with each other's benefits and interests instead of in opposition. This provides safeguarding and protects the project teams' exchanges and interactions across locations.

Finally, I found that when fewer team members interact, the variances that the project team members across locations bring to their interactions, in terms of goals and expectations, are reduced. This leads to less monitoring thus facilitating safeguarding. The following sections set these findings out in more detail.

Restricted access facilitates safeguarding of exchanges in the Lake Project through centralised points of contact. They form a de facto point of contact which allows fewer project team members to be involved in exchanges across locations. The team S business analyst explains *"Centralised points of contacts do save time and monitoring effort because otherwise you would have lots of project team members involved in each interaction across locations."* (BA, team S)

BAs also act as a central point of contact for the core team across locations to communicate their issues. One of the sources of conflict or concern that the Lake Project have is the difficulty the teams have concerning full understanding of what the priorities are beyond what they are doing in the current iteration. BAs and IMs tend to spend a lot of time looking ahead from iteration to iteration while the core the team members are not involved. That is why sometimes the core team members across geographic boundaries are concerned that they are provided with enough understanding and context of the future. A business analyst explains *“This message channel through BAs which prevents seven developers individually having conflict with two BAs on the ground here. So rather than having each developer in China coming to me individually, saying ‘I don’t have enough context about this’, their BA comes to me and says ‘how we can manage that’.”* (BA, team C)

Here BAs are the contact point for the core team members across locations to resolve the issues. It’s the same case with testers. *“From an IM perspective where I tend to hear about issues and conflict or concerns that the testers as a whole have, it will go through to the lead tester. So it’s not like there are five testers coming to me and saying we’ve got a problem.”* (IM, Lake Project)

Another example where the Lake Project structure restricts access for certain types of exchanges is where the core team members across all locations rarely have interaction with the program sponsor or project manager. They communicate at a higher level through the IM as their point of contact. At the social level, the project team members have interactions with any other team members on the project, but at project level these interactions are more restricted to the point of contact for that specific exchange. Depending on the type of exchange, the contact point, the IM, would interact with the project team members. *“When they need [core team members] they come to me and ask me. It depends on the type of problem, I talk with the right person, could be the program sponsor, product owner or project manager.”* (IM, Lake Project) This provides safeguard and protection in interactions and exchanges among the project teams across locations.

These examples indicate the existence of a centralised point of contact in the Lake Project structure, meaning more focussed interactions with fewer project team members involved at any point in time. This decreases the required monitoring of the project teams’ exchanges locally and across geographic boundaries, thus, reducing the possibility of opportunistic behaviours occurring.

As discussed previously, the Lake Project teams formed specific forums for certain project roles across locations. These forums are very critical in the Lake Project *"Because we are distributed, forums are quite important for us to make sure everyone is on one page"* (Developer, team S). The testers, developers and BAs have Testers' Forums, Developers' Forums and BAs' Forums to catch up on a daily basis across locations. Each forum is only accessible by the team members of that forum based on their role in the project. For example only developers can access the Developers' Forum and interact and exchange knowledge. The same applies to other forums. This example indicates there are fewer team members involved in the forums' interactions across geographic boundaries, therefore there is less monitoring required and interactions are more protected and safeguarded. Furthermore, the Lake Project safeguards the exchanges with the customer, it protects interactions between the business side and the core team by restricting access to the business representatives only.

The restricted access is setup so that only the product owner and e-commerce interact with the customer. They act on behalf of the client and work with the project teams as contact points from the business side. The product owner and e-commerce manage the relationship with the customer while the BA represents the business within the core team. The Lake Project IM explains that *"If the core team needs to interact with the customer, they will get the product owner. The product owner and e-commerce are real representatives to the customer for us."*

Consequently the project core team members like developers and testers never interact with the client directly. It is explained that "*Generally speaking, testers and developers should not be conversing with the client over any kind of requirements. It's because of the way we work and being a highly regulated body, we don't actually show anything to the real customers until we have something that we are happy to show."* (Team leader, Lake Project) This example indicates that the Lake Project limited number of team members who can have frequent and continuous interactions with the customer and reduced the amount of monitoring required for these type of exchanges across locations. It means they can focus on the exchange with project outsiders and the transactions between business and the core team are safeguarded as it allows for *"Tight control of the requirements, it keeps everything organised. There's no chance of communication failure."* (Developer, team S)

For security reasons and to protect the Ocean Group live system and customer data, team C, as partner team members, have restricted access to certain data. There are two parts to it; one is data privacy where the Ocean Group restricts team C's access customer data for privacy reasons *"[We are] protecting our customer data, everything that has customer data, China team cannot get access to it."* (BA, team S) The other one is getting access to the Ocean Group technology. Team S and team M are both from the Ocean Group and have consistent access for their interactions on their live system, but team C as partner team has restricted access to this part of the system. It is part of the Ocean Group's security policy to protect their technology and intellectual property. This applies to all projects including the Lake Project. The project manager explains *"There is some restricted access, absolutely. For example, we are writing a new business to our policy system that China does not have access to due to data protection. The nature of the agreement we have doesn't allow that."* (Project manager, Lake Project) This example indicates the Lake Project reduced the number of project team members involved in this type of interaction, thus easing and reducing the monitoring of exchanges that leads to safeguarding the interactions across locations.

Another indicator that provides protection for the Lake Project teams' interactions across locations is embeddedness. As discussed before, there is a strong tie among the project teams across geographic boundaries. The existence of such strong embeddedness reduces the incentives for opportunism because the team members involved in exchanges tend to see their interests and needs as aligned rather than in opposition. It encourages trust and reciprocity between the project team members which also protects interactions and reduces the likelihood of opportunistic behaviours. In addition, the strong embeddedness among the teams across locations makes it possible to have information about the other team members so the team members know with whom to exchange and whom to avoid.

The relational contract is another indicator that protects the Lake Project teams from possible behavioural uncertainties. As discussed previously, the project teams across locations respect the goals and values outlined in team relational contracts and do their best to follow them as expected. They have their relational contract in place which says what the expected behaviours are. *"For our team, this is an ethical contract."* (IM, Lake Project) Being a member of a relational contract reduces the likelihood of opportunism. A business analyst indicates that *"It's about the* *team and what the team goals are and to get aligned with that. So usually when it comes to opportunistic behaviour, it's about not being aligned with the goals of the team."* (BA, team C)

The above discussion indicates that the Lake Project restricts access of the project teams in different transactions across locations. This allows fewer team members' involvement in each exchange and results in fewer monitoring actions. Therefore, the possible opportunism occurrences are decreased.

Macroculture's impact on safeguarding

I found macroculture facilitates safeguarding of the Lake Project teams' exchanges across locations through shared common norms and value. Shared norms like shadower role, buddy role, shoulder check, voicing out issues and avoiding the blame game plus sharing strong trust value facilitate protection for the teams' transactions across all locations. The following paragraphs provide more detail.

I found that in the Lake Project, macroculture facilitates safeguarding of exchanges among the Lake Project teams across geographic boundaries (see Figure 5.1) through (1) shared norms including shoulder check, shadower role, buddy role, avoiding the 'blame game' and ease of being able to voice out issues, and (2) shared values including trust. These shared norms and values across locations reduce the amount of monitoring required and safeguard the teams' interactions across geographic boundaries. One of the shared norms between the Lake Project members is continuous shoulder checks. I found that the constant shoulder check of tasks between developers, BAs and testers before developing the story, after development and before the testing phase safeguards transactions between the project teams across geographic boundaries. I found that such continual checking decreases the chance and likelihood of any opportunistic behaviour to occur.

In addition, the Lake Project teams shared two norms of buddy and shadower across locations. If a team member leaves or changes during the project life, I found that buddy role and shadower role not only ease the work among the project team members but also protect them from opportunism. During the buddy process, new team members start working on a task while they have a buddy. A buddy is usually a senior team member that sits next to a new team member. They usually integrate and get an understanding of what's happening really quickly so they can start bringing value and new ideas to the team. A buddy is there to provide support to new team members and check on them daily.

Another approach is shadowing. In this approach, a new team member (shadower) follows one of the project team members dedicated to him (shadow) around to learn what he does day to day. It's a kind of learning by watching. Then after one week they switch, the shadow person will pull back and the shadower starts doing the work and when the shadow is happy with shadower's job, they back off. I found that both shared norms, the shadow role and the buddy role, decrease monitoring of interactions among the project teams across geographic boundaries and decrease the possibility of opportunism and behavioural uncertainty. Furthermore, I found the shared norm of avoiding the blame game between the Lake Project teams takes away the opportunity of the team members to blame others. This allows a safe environment for the teams across locations and safeguards them from possible behavioural uncertainties.

Finally, the shared norm of ease of being able to voice out issues encourages a 'safe-to-fail' environment for all the project teams across locations so they won't have fear of lock in. They speak out about any issues or problems instead of holding them back. When the project team members have no fear to discuss issues, the issues would not be hold back until it is too late, so the issues become much more transparent and easier to handle. It is especially critical to have them in distributed agile development project teams such as the Lake Project as part of their safety features as they reduce the possibility of opportunistic behaviour. The following sections set these findings out in more detail.

1. Shared norms

These shared norms are explained in details previously. Shoulder check or peer review is done pretty constantly across the team to check the tasks with developers, testers and the business. It starts before a developer picks up a story to develop and continues all the way when it goes to the testers and to business. The teams across locations are continuously communicating and checking if the task is communicated properly when it is passed to another team member for the next phase. This constant communication and checking across the project on each task provides a safe environment and protects interactions across locations. Any issues like lack of context, ambiguity or misunderstanding is uncovered immediately and not left for later.

Shadower roles are an additional source for the role they shadow (the shadower) in the Lake Project. They are back up roles to replace the project members when they go on leave or change during the project. The shadower role safeguards the project team and provides better flexibility if a team member leaves the project suddenly. In this situation, there is always an extra person to come in and fill that role straight away because they have been there watching how things are done. When a shadower fills the missing role, it provides more safeguarding to the project compared to when the role is filled by a new team member outside the project team. The shadower role lessens monitoring requirements and protects the project teams' interactions across locations because the shadower learned the team norms, values, goals and working culture and this reduces the possibility of becoming a victim of opportunistic behaviour.

If a project team member leaves the project and there is no shadower role to fill the role the project has to recruit a new team member for the project. In this case there is another safeguard practice used by the Lake Project called the buddy role. The buddy is usually a senior team member that sits beside the new team member and takes him/her through step by step, makes sure they are familiarised with the context, the norms, values and goals of the project. Buddies act as a support and checks on the new team member on daily basis until the new team member is well enough to work independently. A developer explains *"A lot of the times we tried to pair the new member up, so we find a buddy for them to introduce them to the project. You know it was really around let's giving this person a piece to be responsible for and then guide them and support them through."* (Developer, team S) The Lake Project protects and safeguards the teams' transactions across locations with this practice and leaves a low level of opportunity for inappropriate behaviours.

Avoiding the blame game, as explained before, is a shared norm between the Lake Project teams across locations. If an issue is raised by a team member, that team member cannot be blamed and or confronted in a negative way. The focus is more on resolving the issue and fixing the problem, not searching for whose fault it is. There is a shared understanding among the teams that there is always a chance for improving and making things work better for the whole team otherwise there would be a lot of conflicts among the teams, especially across locations. This shared practice provides safeguarding among the project teams because the team members, instead of keeping the problems within and become stressed to communicate them with the other team members, they feel safe to openly talk about issues as early as possible without having the fear of being blamed. Thus, it facilitates a 'safe-to-fail' environment which decreases the likelihood of possible conflicts or opportunistic behaviours in all locations.

Another common strategy that is encouraged extensively among the project teams in all locations is to voice out any issues or risks as soon as you can, before it's too late. As the Lake Project team leader would say *"bad news doesn't get better with age"*. As explained previously, the teams are encouraged to voice out and raise issues during the daily Stand ups, Retrospectives and Showcase meetings or any other team socialisation sessions and communicate them through to their managers quickly. This practice is well communicated to all the project teams across geographic boundaries because the Lake Project members frequently witness the other team members feel safe to voice out risks, issues and their opinion through the project life. This helps in having an overall safe and trusting environment and protection to the teams' interactions across locations. It also reduces the possibility of opportunistic behaviours across geographic boundaries.

Having shared understandings and norms like avoiding the blame game and being able to voice out issues among the teams enhances the conflict resolution strategy between the teams across geographic boundaries. When there is no option to assign the blame to a team member, in a way it protects the teams from conflicts, or if a conflict is raised it can be collectively resolved by the teams instead of holding back for too long. When team members voice out their issues and concerns, the other team members help to resolve the problem. For example, if the problem is conflict among the team members or a disagreement on a particular subject, the other team members collaborate and work on it to resolve the raised conflict. Indeed, a result of that, the conflict is resolved as part of trying to solve the problem. The Lake Project IM explains that *"[Having] a kind of safe space within the team and to be able to say that something that is not working or you don't feel like you are getting value out of particular activity or a process and having that norm that there is no blame attached to that... you are kind of seeking how to improve the process for the team rather than find out whose fault it is. It prevents conflict in the team."* (IM, Lake Project)

2. Shared values

Shared common values among the project teams across all locations facilitates a safe and trusting work environment for the Lake Project teams. Trust is a strong value among the teams in all locations and they "*have been living that value very much*" (Developer, team S). This is very evident through examples demonstrated in section 4.7.1. In one of the examples, team C had to follow team S to complete an integration of a big piece of work. Considering team C is a partner team that is located in China, it was a very challenging job to do. However, the Lake Project teams were able to do the job properly because a high level of trust existed among the teams across locations.

One of the reasons for having such a trusted working environment in the Lake Project is that trust is not only part of the project team's values but also part of the Ocean Group and the River organisational values. Therefore, the project teams' working culture was based on trust even before joining together for this project. Both companies encouraged and reminded their staff on a regular basis so it becomes part of their working culture and they brought this value to the Lake Project. In addition, practices such as co-locating the project teams at the early stages of the project and through the project life cycle promoted trust among the teams. Besides that, shared norms like avoiding the blame game and being able to voice out issues at any time encouraged celebrating failures and creating a trusting and safe environment. This enhanced trust among the teams and decreased the risk of opportunism in the Lake Project.

The above discussion indicates that shared values like trust among the team members across geographic boundaries facilitated a safe and trusted environment for all the project teams. Therefore it decreases the incentives for the project teams to become victims of opportunistic behaviours.

Collective sanctions' impact on safeguarding

I found the occurrence of opportunism and unacceptable behaviour was extremely low in the Lake Project team. The reason for this is because they are using the right practices to encourage the visibility, positivity and trust in the project across locations. The following paragraphs provide more detail.

I found in the Lake Project that collective sanctions safeguard the project teams' interactions through a gradual escalation system, motivating positive attitudes and behaviours across geographic boundaries, encouraging a voice out norm to facilitate 'safe-to-fail' environment and finally through using meta-norm and peer pressure practices. All of these practices reduce monitoring effort and cost, thus decreasing the behavioural uncertainty across all locations.

The gradual escalation system escalates unacceptable behaviours of the teams to the higher levels gradually. It starts with talking through the behaviour with the team member, finding out the reason and a reminder of the consequences if the behaviour is repeated. The team member is referred to higher level if the behaviour continues. If the escalation continues, the team member may face financial consequences or total exclusion from the project. This system has the benefits of giving a second chance to the team members to correct their behaviour and make them feel they are still part of the team. It also prevents any blaming condition and unnecessary sanctions if there has been a misunderstanding about the behaviour. I found that this attitude maintains the trusting environment among the project teams across locations and protects their interactions.

Collective sanctions safeguard interactions and exchanges among the Lake Project teams across geographic boundaries by encouraging the teams to act in a manner that is regarded as competent and professional and to behave in line with the project expectations. The project team members are encouraged through informal incentives such as talking highly of high performers or showing them recognition of acceptable behaviours. The Lake Project identified the acceptable behaviour parameters through their values, norms and relational contracts. If an unacceptable behaviour happens that is in contrast with the team relational contract and the Ocean Group values, the team members in all locations raise it as a concern.

The goal is to encourage the other project teams across locations to use the rewarded behaviour as their role model and get motivated to behave the same in order to gain positive reputation, be regarded as a professional team member and gain trust of the other team members. I found that by using this practice, the Lake Project safeguard and reduce the likelihood of unacceptable behaviours. In addition, I found that the Lake Project teams across geographic boundaries are encouraged to voice out issues and problems in order to increase the visibility of the problems in the earlier stages thus reducing monitoring costs and behavioural uncertainty across locations.

Furthermore, meta-norm and peer pressure practices were found to be effective in protecting the teams' opportunistic behaviours across all locations. In meta-norm practice, if a senior team member or a manager does not monitor or act when an unacceptable behaviour occurs, then the senior team member will face consequences. This encourages the senior team member to constantly monitor and act against unacceptable behaviours. Similarly, in peer pressure practice the project team members are encouraged to informally monitor other team members' activities to decrease opportunism. I found it is usually more visible during pair programming, shoulder check and code review practices across geographic boundaries. I found that meta-norm and peer pressure both decrease monitoring effort and cost, thus decreasing opportunistic behaviours across locations. The following sections set these findings out in more detail.

In general, the Lake Project does not believe in imposing severe punishments and financial consequences on the project team members straight away but they act in a softer way. They believe imposing negative sanctions may have short term effect by it does not bring long term protection as much as when the team members feel they are part of the team. For example, the Lake Project team leader explains that *"Financial consequences are every 12 months and it can be off the radar as we go through the review process. So you are not thinking what I am going to do today to lose or gain bonuses in the middle of the next year. It's more about the bond with the team. When you have a common goal and have a joy of working with the team to share the outcome, people mostly get out of it."* (Team leader, Lake Project).

One of the main reasons that the Lake Project is not willing to impose severe negative sanctions is that they do not want to damage the strong bond and trusting working environment that exists among the teams across geographic boundaries. Instead they use encouragement and incentives to enhance the trusting environment among the team members, so team members become more motivated and feel they are part of the team. Therefore they use practices like a gradual escalation system and encouraging positive and well regarded behaviours. Thus, in this way they promote a safe working place and protect the project teams' exchanges across locations.

As discussed in collective sanctions, if a team member behaves unacceptably or is violating norms, goals or values of the project team, the Lake Project follow a gradual escalation system. The first step is to talk through and find out the reason for such behaviour. This helps the teams to be protected from any blaming and the sanction will not be imposed on them based on a misunderstanding, because sometimes there are genuine reasons behind the behaviour. Majority of the time, this practice works well. If not, the matter is escalated to the team leader where he also talks with the team member and tries to remind him of the values and the effects of his behaviour on the project team. The team member is also reminded of the consequences of his behaviour if repeated. The gradual escalation continues and if the unacceptable behaviour is repeated frequently or in severe cases, if it damages the project progress, they have to exclude the team member from the project.

In the Lake Project, the teams are encouraged through informal incentives such as talking highly about high performers or show them recognition of acceptable behaviours. A developer explains *"There is an avenue we can raise it which is the Retrospective that happens every two weeks or we can actually raise it from day to day. A lot of time it will be raised from day to day. We talk about it and if we need to change the relational contract we change it."* (Developer, team S)

The positive actions are encouraged, mentioned casually, turned around and talked about. The Lake Project team leader says *"We acknowledge positive behaviours during Retrospectives and thank team members for positive attitude. This becomes a norm in our regular exchanges across locations that negative attitudes or comments are stopped by team members."* The Lake Project IM adds *"When they do things, we recognise and appreciate that they've done things well, that they shared something that helped us."*

The motive behind these actions is to push that example out there as a role model for the less engaging team members so hopefully they are tempted to adopt and get engaged with those behaviours more often. The reason for their motivation is that the acceptable behaviour or high performance is regarded as a very positive action by the team members across locations and gains their trust and respect. A business analyst believes that the project team members follow their leader's behaviour *"It all starts from you, team members look up to their tech leads or their BA leads, if you are more settled you always model the right behaviour. If the people messed up time after time you don't yell at them, you express it in another way."* (BA, team S) This means as much as the lead roles can be the model for right behaviour they also can be the model for wrong behaviour.

The Lake Project strategy is you cannot expect the teams to behave in certain way if you do not act in line with it too. If a team member is acting in line with the norms and values they are expected to, it *"Shows everyone you are doing it and makes it very visible that you are doing it. In this way they become more transparent in their daily activities too."* (BA, team C) Encouraging these practices across the project team helps to safeguard teams from possible inappropriate and opportunistic behaviour.

In addition, the Lake Project teams are encouraged to act in line with the expected behaviour by increasing the visibility of what the project expectations are and what the right path is. The teams are encouraged to be on that path by giving them a choice about what they want to do. In other words, the Lake Project gives the team members a vision of what the project wants to achieve. If team members are not or cannot be aligned with that vision, the Lake Project ask them to leave the project *"[We] want them to go and look somewhere where they feel happy. Once again you give them a sense of direction; you give them visibility of what the view is."* (Project manager, Lake Project)

Furthermore, the Lake Project uses 'voice out' practice to increase the visibility of the project teams' actions across geographic boundaries. The Lake Project encourages the project team members to speak up about their issues in social sessions like daily Stand ups. The intention behind this practice is to make the project team member feel 'safe-to-fail', encourage them to express issues and make it clear when they have failed or exhibited a negative behaviour. The Lake Project IM also adds *"We have a guy in China go and say 'hey guys I've done this and I really screwed up and it's not going to work and you know I'm sorry' and everyone goes 'well let's correct it and let's do not do that again'. So we say let's celebrate failure and make sure that they are not locked in and isolated."*

This example indicates that the Lake Project is providing a 'safe-to-fail' working environment for all the teams across geographic boundaries by encouraging speaking up and sharing the wrong behaviour. This 'safe-to-fail' environment reduces the amount of monitoring required in all locations and increases trust among the team members thus protecting the teams from possible opportunistic behaviours.

One of the practices used in the Lake Project across locations is meta-norm, where a senior project team member must act when a team member does not behave appropriately according to the team‘s norms, goals and values. For example, if a lead BA, IM, team leader or the project manager does not monitor low performers in the project their own reputation will be at risk. Being a right model for the other team members across locations by always behaving in professional and acceptable way protects the leader roles in the project from consequences.

Therefore, meta-norm is an incentive for lead roles to act according to their responsibilities and feel accountable for unacceptable behaviours of the team members. Otherwise, *“We are questioned why we did not take care of the situation, why we did not monitor the case and did not report it on time.”* (IM, Lake Project). Another lead member of the project also adds *“As a lead you should be looking at monitoring inappropriate behaviours and you should be looking at having a more positive attitude.”* (Team leader, Lake Project) This encourages team members to act competently and professionally across all locations and reduces the possibility of becoming victim of opportunistic behaviours.

Peer pressure is another practice that is used by the Lake Project team members as a safeguard for wrong behaviours, where the team members informally monitor the other team members locally and across locations. This monitoring practice is informal that increases the potential of good behaviours and brings more discipline within the team. For example, every team member is watching or following the other team members’ activities on daily basis as part of their daily activities in order to be updated. If they notice something is going wrong they voice it out. A business analyst gives an example: *“I noticed that in a co-located office it is easier because you see things. For example, I noticed somebody struggling bit, I go ‘is everything all right? What can I do?’ I ask out of concern. So we have peer pressure to raise underperformance and peer questioning of the individual.”* (BA, team S)

The informal monitoring is often done when the project team members pair program across geographic boundaries. During pair programming, there is a team member who is monitoring the other team members’ work and if they notice inappropriate behaviour, having in mind the existing safe working environment, they raise it as a concern. The same applies to shoulder check and code review practice, *“In this way we manage to have the team much more disciplined and behaviours are more aligned with the Lake Project expectations.”* (BA, team C) The above discussion indicates collective sanctions use different practices that facilitate safeguarding for the project teams’ exchanges locally and across geographic boundaries.

Reputation's impact on safeguarding

As discussed, I found that it is important for the Lake Project team members to gain and maintain their positive reputation. When reputation is perceived as important, individuals will be motivated to protect their reputations by avoiding deceptive or self-interested behaviour. Therefore, the underlying driver of safeguarding in the Lake Project is the desire of the project team members to protect their reputations.

I found reputation facilitates the safeguarding of exchanges in similar way to how it facilitates coordinating exchanges across borders. Reputation facilitates the Lake Project teams' interactions across locations through practices like resourcing the project from high reputation team members, the high profile and importance of the Lake Project, detecting and diffusing the team members' behaviours, high visibility of team members' actions and finally the existence of a high level of trust among the team members across geographic boundaries. These practices encourage the project team members to maintain their positive reputation thus providing less motivation for behavioural uncertainty and opportunisms. The following sections set these findings out in more detail.

In addition, a motivation for the Lake Project team members across locations to present a positive reputation is the importance of the Lake Project within the company and also in the wider network. Being a high profile project, the Lake Project itself is an incentive to demonstrate positive reputation. It has been discussed that it is important for the Lake Project team members to protect their reputation for different reasons, for example, majority of the team members were already skilled and highly regarded before joining the Lake Project. Because the Lake Project is a high profile project, being part of the Lake Project team is regarded as an excellent opportunity by the project team members and can be used as a reference in their work experience for future work opportunities and offers.

Therefore, there is incentive for them to not only protect their existing reputation but it is also a great opportunity to build an even better reputation. Thus, having the team members with a positive reputation based on their previous quality of work and being a team member of a high profile project decreases the incentives for unacceptable and opportunistic behaviour. In other words, it is a way that the Lake Project can safeguard and protect the future interactions among the team members and reduce the occurrence of opportunistic behaviours as much as possible.

Another practice that motivates the team members to have positive reputations is the high visibility of the project. When the team members see their actions are visible to the other team members, this motivates them to act as competently as possible. So, the behavioural uncertainty is reduced and the team members' transactions are protected.

In addition to the above practices, I found that strong embeddedness among the Lake Project teams facilitates sharing information about the team members' reputation and their behaviours throughout the entire team locally and across locations. This flow of information is done through word of mouth and frequent role rotations in the team. There is quite regular process of peer switching within the team. *"Project members switch around so much, this makes it much easier to either know other team members or there is always somebody that you can easily get the information about their reputation from"* (Developer, team S). When a team member asks or hears about another team member's reputation and does not hear many positive comments, it is another way of not presenting a positive reputation to others.

One of the important indicators that protects the teams' interactions across locations is the existence of trust in the team members' ability to perform tasks based on the previous quality of their work. Therefore, the team members make positive reputations based on trust. The distributed nature of the project makes it challenging to complete tasks across geographic boundaries, but the existence of trust based transactions between the teams across locations reduced the amount of monitoring required. Thus, trust based reputation protects the teams from being a victim of opportunistic behaviours.

5.5 Summary of Results

This chapter presented results of the data collection and analysis conducted for this exploratory study. Information was presented on how participants were recruited. The analytical methodology was described as well as the resulting themes that emerged from the data. Selected quotes were presented for each of the lower-level themes that described their stories and experiences in response to the research questions.

The problem addressed in the study was ‘how can distributed agile development projects be governed effectively?’ I used NG theory as a theoretical lens to identify and understand distributed agile development project governance mechanisms, which led to emerging research questions. Participants included the Lake Project team members from the project level and the core team level. The participating team members were from the Sydney team as well as the China team. The data were recorded, coded, reviewed and analysed.

I found clear evidence for existence of all four social governance mechanisms including restricted access, macroculture, collective sanctions and reputation in the Lake Project. The analysis demonstrated that all four social mechanisms can resolve exchange problems through coordinating and safeguarding of the project team exchanges across locations. Finally, from our analysis, it is apparent that social governance mechanisms enhance coordination of distributed agile development project teams and protect their interactions and exchanges across locations. The summary of findings for the research question 1, 2 and 3 are presented in Table 5.4.

Table 5. 4 Summary of Social Mechanisms Resolving Exchange Problems in Distributed Agile Development Projects

Social Mechanisms (Jones et al. 1997)	Constructs Confirmed (Jones et al. 1997)	Application in the Case	Facilitate coordination	Facilitate Safeguarding
Restricted access	Status Maximisation	Similar status members interactions more frequently	High interaction frequency	Frequency of interactions (strong embeddedness)
	Embeddedness	Fewer members interact more frequently	High interaction frequency	Restrict access to certain members for certain type of exchange
	Ideological Similarities (New construct found)	Members with similar interests interacts more frequently	High clarity of roles	Centralized point of contacts
	Relational Contracting	Membership motivates fewer member to interact		Membership in relational contract
Macroculture	Shared norms	Shared common agile practices Shared non-agile practices Shared common working standards Shared common strategies Shared common approaches Shared common language	Shared norms to prevent fear of lock-in Shared common language Shared common approaches and solutions to particular situation Shared common agile practices	Create a 'safe-to-fail' environment through shared norms
	Shared values	Ocean Group values River values Agile values Relational contract values	Unified team members' expectations across geographic boundaries	Create a 'safe-to-fail' environment through shared values such as trust
	Shared goals	Shared goal of delivering the project on time Shared goal of delivering to the project technical strategy Shared focus on delivering software to the satisfaction of the business Shared goal of making the distributed agile component work	Shared focus on delivering software to the satisfaction of the business	
	Shared assumptions	To communicate the same message to members and the business To work agile Not to work on weekends and until late hours Either participate in all team activities or challenge them but cannot withdraw from them To discuss issues comes up during iteration with lead BA or IM To monitor blockers on the wall	Encourage new members to be compatible with project culture	
Collective sanctions	Actions to condemn the unacceptable behaviour	Financial consequences (i.e. losing bonus)	Organise the project team members around agile values	Organise the project team members around agile values
		Non-financial consequences (i.e. assign different task, gradual escalation, spreading words for misbehaviour)	Encourage the team members to act in line with acceptable behaviours through high level of transparency and visibility of the team members' activities	Encourage the team members to act in line with acceptable behaviours through high level of transparency and visibility of the team members' activities
	Actions to reward the acceptable behaviour (New construct found)	Formal rewarding actions (i.e. financial bonus , Thank You System)		
		Informal rewarding actions (i.e. words of encouragement)	Encourage team members to behave in a manner that is regarded as competent and professional across borders through shared norms and reward acceptable behaviours	Encourage team members to behave in a manner that is regarded as competent and professional across borders through shared norms and reward acceptable behaviours
Reputation	the perception that project members hold about another member's character, knowledge and skills	Social expectations: Importance of fulfilling social expectations	Resource the project with members demonstrated positive reputation and competency Detect and diffuse team members' behaviours High profile and importance of the project acts as an incentive to protect reputation	Resource project with members demonstrated positive reputation and competency Detect and diffusing team members' behaviours High profile and importance of the project acts as an incentive to protect reputation
		Individual expectations: Importance of fulfilling individual expectations	High level of trust between team members High level of visibility of team members' activities	High level of trust between team members High level of visibility of team members' activities

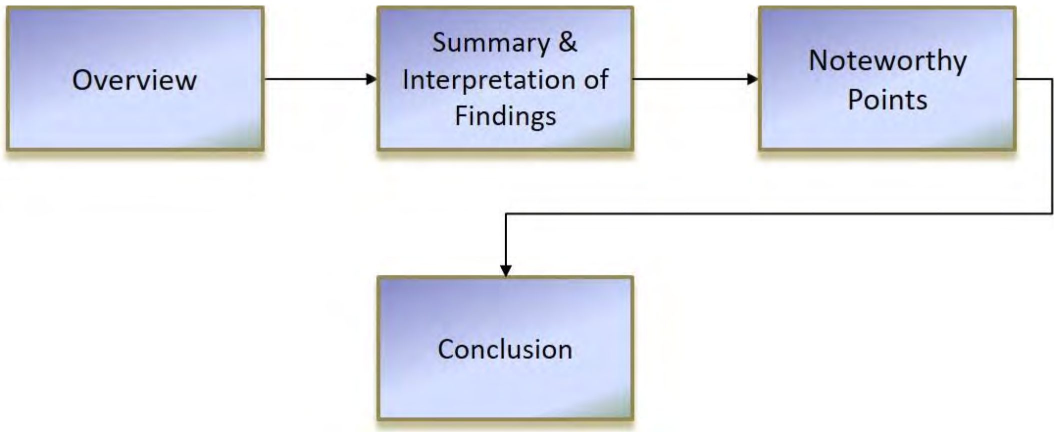
Chapter 6 contains an interpretation of the research findings. Then, Chapter 7 includes an explanation of the limitations of the study, recommendations for practitioners and suggestions for future research. The chapter concludes with a discussion of implications.

Chapter 6 Discussion

6.1 Overview

Chapter 4 included a report of how the data analysis was performed and Chapter 5 included the result of the study. This chapter discusses the summary of the main findings, interprets and explains what the results mean and discusses how the results relate to the literature in light of previous research (confirmed or refused previous studies) by comparing it with other research. It also provides answers to the current research questions and discusses the significance and importance of the results. Subsequently, the chapter provided an additional discussion about the research's noteworthy points of what appeared to be useful for the practitioners. Finally, the chapter closes in a conclusion section summarising the findings and discussion (Figure 6.1). These findings enable understanding in how social governance mechanisms are applied in distributed agile development projects and what their impacts are on coordination and safeguarding of exchanges. This is discussed in detail in the following sections:

Figure 6. 1 The Structure of Chapter 6



6.2 Introduction

The current research began by advocating that distributed agile development projects can be conceptualised using social governance mechanisms. This is because the essential characteristics of social structures where developers work in a complex and uncertain environment are present in distributed agile development projects. In addition, the exchange conditions with high coordination, high adoption and high safeguarding exist in distributed agile development projects.

During the previous two decades only a few reported experiences of applying distributed agile development to industrial projects can be found (Ramesh *et al.* 2006, Sutherland *et al.* 2007, Paasivaara *et al.* 2008), with even fewer case studies available in the literature (Paasivaara *et al.* 2009, Korkala & Abrahamsson 2007). As distributed agile development becomes more common place and widely adopted, many organisations are interested in bringing distributed agile development into use or, in some cases, have already started to use it (Sureshchandra & Shrinivasavadhani 2008). However, there has not been a significant effort in literature to suggest any mechanisms as the solution to address the distributed agile development challenges such as coordination and safeguarding problems. This study aimed to cover this gap in literature.

The purpose of this qualitative study was to explore the effective governance mechanisms being applied in distributed agile development projects and to understand their impact on coordinating and safeguarding exchanges in distributed agile development teams. All four research questions related to the framework, as discussed in Chapters 3 and 5.

In summary, I found distributed agile development projects could be conceptualised by applying NG social mechanisms to coordinate activities and safeguard exchanges. The findings are consistent with the NG literature in regards to the fundamental application of social governance mechanisms. However, I also added new findings and prospects to the previous case studies. For example, applying Jones *et al.* (1997) NG framework in distributed agile development projects demonstrated that social governance mechanisms are context dependant meaning they work differently in various contexts which results in having different impacts on coordinating and safeguarding exchanges.

Furthermore, the results of this study confirmed all of the previous relationships in the Jones *et al.* (1997) model and previous literature in distributed contexts as follows:

- Restricted access coordinates the project teams' activities (Jones *et al.* 1997, Sagers 2004, Feller *et al.* 2008)
- Restricted access safeguards the project teams' interactions (Jones *et al.* 1997, Sagers 2004)

- Macroculture coordinates the teams' activities (Jones *et al.* 1997, Feller *et al.* 2008)
- Macroculture safeguards the project teams' interactions (Feller *et al.* 2008)
- Collective sanctions facilitate coordination of the projects team members' activities (Feller *et al.* 2008)
- Collective sanctions safeguard the project teams' interactions (Jones *et al.* 1997, Sagers 2004, Feller *et al.* 2008)
- Reputation safeguards the project teams' interactions (Jones *et al.* 1997, Sagers 2004, Feller *et al.* 2008)

In addition to the above relationships, I found the model was incomplete and one new relationship emerged as follow:

- Reputation facilitates coordination of the project teams' activities across geographical boundaries.

One of the new relationships found, was that reputation can facilitate coordination of the project teams' activities across locations in the same way it safeguards the teams' exchanges across geographical boundaries. Another finding was the application of the four social mechanisms in distributed agile development concept, which was different compared to their application into other contexts in previous studies (Jones *et al.* 1997, Feller *et al.* 2008, Sager 2004). One of the main findings of this study was the appearance of the new construct, 'ideological similarity' that emerged by applying restricted access in the Lake Project.

In addition to the above formal research results, the study noted some noteworthy points such as the importance of social mechanisms congruency. It suggests that companies can have more ability resolving exchange problems effectively by developing the right social mechanisms which are concurrent with each other, in particular with prevailing macroculture. Finally, as parts of the noteworthy points, the study noted potential interactions amongst the four social governance mechanisms indicated the critical role of macroculture, its impact on other social mechanisms applications and the way they resolve exchange problems. These points are discussed in more details in section 6.8.

The following sections discuss the social mechanisms in turn. They provide briefs about each mechanism's concepts and their impact on coordination and safeguarding exchanges across locations, which are the formal results of the current research.

6.3 Restricted Access

In the Lake Project, the number of the team members that could potentially interact with each other across locations was relatively large. By significantly restricting the number of team members that collaborate to produce the software, coordination was enhanced and the uncertainty of future exchanges was reduced. A new construct that contributed and emerged through the findings was 'ideological similarity'. This section first talks about the restricted access concept and 'ideological similarity' as a new construct, and then explains the restricted access impact on coordinating and safeguarding the project teams' exchanges across locations.

6.3.1 Restricted Access Concept

Restricted access is about reducing the number of team members involved in certain types of exchanges across the project team. For example, there are limited interactions between developers and the customer in the Lake Project. If the developers need to interact with the customer, they do so via the product owner and the e-commerce representatives. Such channelled interaction with fewer team members not only makes it easier to communicate and interact but it also reduces the risk of opportunistic behaviours occurring, as the Lake Project team leader explains, *"For example, if developers interact with the customer on frequent basis, because they* [developers] *are on the technical side of the interaction and the customer is on the business side, there will be a lot of miscommunications and confusion involved. But having the product owner and the e-commerce representatives as the mediator reduce the risk."*

As discussed in Chapter 5, restricted access is operated through different constructs including status maximisation (Jones *et al.* 1997, Podolny 1994), relational contracting (Williamson 1985, 1991, Bolton *et al.* 1994, Jones *et al.* 1997, MacNeil 1980), and embeddedness (Granovetter

1992, Jones *et al.* 1997). In addition to these constructs, I found a new construct 'ideological similarity'. Indeed, I found that all of these constructs facilitate focussed exchange amongst fewer team members in the project that results in a high level of contact and interactions amongst limited team members based on their status (status maximisation), commitment (relational contracting), interests (ideological similarity) and strong ties (embeddedness) amongst them.

The empirical results of the case study showed that the Lake Project team members preferred to interact with the other team members of similar status both locally and across locations on a regular basis. These interactions reduced confusion and saved time and effort because it made the project team members interact with fewer team members. These findings are consistent with the argument in the NG literature where status maximisation strategy restricts access because the project team members seek to avoid the other team members of different (lower or higher) status; therefore the result of status maximisation is exchange amongst team members of similar status (Jones *et al.* 1997 and Podolny 1994).

The Lake Project facilitated a status maximisation strategy for the project teams by structuring the project into different levels (program/project level and core team level), having forums for similar status members (BA forum, tech lead forum, developers forum), and meetings for similar status members such as 'Showcase' meetings at the program/project level and 'Homemade Jam' meetings at the core team level. The similar status forums and meetings and project structure made it easier for the project team to interact with similar status members across geographic boundaries. Therefore, by limiting the exchange amongst the team members with different status, the Lake Project teams were able to reduce the confusion, having less complicated interactions and improve the coordination especially across geographic boundaries, otherwise coordination would be challenging.

This finding agrees with literature (Jones *et al.* 1997) but it also extends the concept in a way that the strategy that firms use to facilitate the status maximisation needs to be carefully chosen not to have conflict with the current values and goals of the project (macroculture). For example, although the Lake Project was structured into different levels of program/project level and core team level but the team members were still able to interact with each other from different levels when they needed. This shows the Lake Project restricting access strategy compatibility with the macroculture (agility) culture and value of the project. Therefore, I found this concept is actually quite deeper than expected and requires proper understanding of the project macroculture. For example, agile values and norms were part of the Lake Project macroculture and having specific forums and meetings for similar status members did not stop the Lake Project team members interacting with other status members if they wished to.

Relational contracting was another construct that applied restricted access amongst the Lake Project teams across geographic boundaries. In the literature, a relational contract is referred to as bilateral or hybrid governance structures by Williamson (1985, 1991) and as relational contracts by MacNeil (1980) and Jones *et al.* (1997).

In the NG literature, relational contracting restricts access through facilitating more frequent interactions amongst fewer team members (Bolton *et al.* 1994, Helper 1991). Indeed, relational contracting is about bounding the project team members' behaviour. Through this contract, that was exclusive to the project teams only, the Lake Project team members were committed to develop a relationship that enabled them to co-operate and meet other team members' requirements, which in term provided them with long term benefits. This point was captured by BA from team S who states, *"A relational contract is important to us and we are committed to follow that up because it helps the project run smoothly."*

By way of example of this relational contract, one of the values that the Lake Project teams were committed to, through their relational contract (Appendix H) was 'do not leave broken build overnight'. This means the teams across locations were committed to not leaving any task or coding that was not running and had errors until tomorrow. This was very important for all of the teams across the different locations because the next morning when the team members from other locations wanted to continue the finished tasks, if the tasks or coding was broken, they would not be able to continue their tasks and it should be left pending until the other sites are online again. This meant waste of effort, time and cost. Therefore, such commitments by the Lake Project teams had long-term benefits for all of the team members across borders. Similar to the literature (Bolton *et al.* 1994, Jones *et al.* 1997), I found that commitment to relational contract was guided by frequent interactions amongst the Lake Project teams across locations.

One of the key results of the case study was evidence of strong embeddedness among the project teams locally and across locations. Embeddedness is found to be critical in understanding how social mechanisms coordinate and safeguard exchanges in NG (Jones *et al.* 1997), hence it's one of the most important indicators. Embeddedness facilitates coordination and safeguard exchanges of social mechanisms including restricted access, macroculture, collective sanctions, and reputation (Jones *et al.* 1997).

Structural embeddedness makes restricted access possible because it provides information so that the team members know with whom to exchange and whom to avoid (Jones *et al.* 1997). Due to strong ties and embeddedness amongst the Lake Project teams, everyone knows each other and everyone knows about each other's skills, capabilities and personalities. This knowledge and familiarity with the other team members makes restricted access possible because it makes it much easier for the team members to know whom they need to interact with for certain types of exchange. It also safeguards their interactions because, with the knowledge they have from the other team members, they get to know who to avoid interacting with. This focussed interaction with fewer team members not only makes it easier to communicate and interact but it also reduces the risk of opportunistic behaviours to occur. It is especially critical for distributed agile project teams, such as the Lake Project, to have this image of the other team members due to the large number of the team members involved in the project distributed across locations (Kotlarsky & Oshri 2005). It helps to improve the social aspects of the team members such as rapport across borders (Kotlarsky & Oshri 2005).

This finding is consistent with the literature that embeddedness is a driver for spreading values and norms of the team members across the project, which in turn enhances coordination amongst teams (Granovetter 1992, Jones *et al.* 1997), and spreading information about the project team members' behaviours and strategies, which enhances safeguarding customised exchanges (Granovetter 1992). In fact, negative information about the other team members reduces the likelihood of direct interaction while positive feedbacks strengthen the likelihood of direct interactions (Burt & Knez, 1995), which lessens the likelihood of opportunism.

Nevertheless, some studies such as the one conducted by Uzzi (1997), suggest that too much embeddedness may create problems because it can reduce the flow of new information into the network. In this way, there are few or no links to outside team members who can potentially contribute with new ideas and information. In the Lake Project, although the project teams had strong ties across locations, they maintained their links with the other Ocean Group IT project team members through frequent 'Homemade Jam' meetings to update and receive feedback from outside the Lake Project. The study suggests that the Lake Project had a rear optimal level of embeddedness.

Finally, ideological similarity is a new construct that emerged from the current study and was used by the Lake Project to create restricted access. Ideological similarity is about the team members with similar attitudes and beliefs in regards to the project and their professional skills and knowledge. Although restricted access is part of NG but ideological similarities is missing in the NG literature. The case study results showed that the tendency of the Lake Project team members to frequently interact with the other team members that have consistent and similar professional and technical interests, restricts access in the Lake Project. This encouraged the project team members to choose interacting with fewer team members locally and across locations regularly. This was facilitated through forums for different groups of the project team members across locations with similar interests, such as developers' forums, BAs' forums and testers' forums. A developer explains, *"I really like to work with the developer in team C because like myself, he is more interested in developing a specific part of the context. It is much easier when we have similar interests especially when we are coding together" (Developer, Team S).* In addition, the project team members found it easier to have more frequent exchanges with the other team members across geographic boundaries who have similar nature of work, such as front end developers. This made for more frequent interactions amongst the team members that were ideologically aligned across locations.

6.3.2 Restricted Access Impact on Coordination

The results of the case study indicated that restricted access facilitated coordination of exchanges amongst the Lake Project teams across geographic boundaries through high interaction frequency, and high clarity of roles. The following section discusses how restricted access facilitates frequent interactions, why frequent exchange is important and how it enhances coordination and safeguarding exchanges.

The indicators of restricted access discussed in the previous section made less team members interact for certain types of exchange across locations, such as ideological similarities. This new construct makes fewer team members interact more frequently across the locations. According to the team members these interactions helped the teams to save time, facilitate communication and lessen confusion. This is more visible in forums for different groups in the project team, such as the developers' forum.

This leads to a high frequency of interactions amongst certain team members of the Lake Project with similar interests for specific types of exchange. The limited interactions amongst fewer project team members for certain types of exchange saves time, effort and cost of communication, especially for exchanges across geographic boundaries. This is consistent with the argument in literature that fewer project team members increase interaction frequency, which can raise the team members' motivation and ability to coordinate smoothly (Jones *et al.* 1997).

When fewer project team members are involved in exchanges they interact more frequently (Jones *et al.* 1997). The constant and frequent interactions among the project teams enabled them to get to know each other's expectations and become familiar with the other team members' skills, goals and their abilities across locations (Bryman *et al.* 1987). In addition, high levels of interaction motivated the Lake Project teams to learn about the other teams' ways of working in each location (Eccles 1981, Faulkner & Anderson 1987), share their knowledge, develop common language and communication protocols and establish routines and similar ways of working across geographic boundaries (Bryman *et al.* 1987). All of these increased the motivation and ability of the project teams to coordinate smoothly across geographic boundaries. This finding agrees with the literature, where Jones *et al.* (1997) argues that restricted access reduces coordination costs by minimising variance in participants' expectations (Feller *et al.* 2008), skills and goals, developing communication protocols and establishing routines from continued interactions.

Frequent exchange amongst the project teams is one of the exchange conditions that determines which governance form is more efficient than the others (Jones *et al.* 1997). In addition, frequent exchange is identified by Williamson (1985) as an important determinant of governance. It is also one of four exchange conditions that determines governance efficiency (Williamson 1991).

In this regard, Williamson (1985) argues that frequency of interactions is important because firstly, it facilitates transferring tacit knowledge; especially where specialised knowledge (human asset specificity) is required. This is compatible with the study finding where, specialised knowledge was required to be transferred to develop the software. Through frequent interactions and contact amongst the project teams across locations, the team members were able to share and transfer technical knowledge. This knowledge deepened through continued interactions. For example, part of the project required integration that was only possible through frequent interactions between team S and team C. The project team members were able to transfer and share their skills and context information through frequent interactions across locations and have a successful integration.

The Lake Project teams were required to have specific knowledge and skills to customise the software throughout the project for customers, in order to develop the required software product. The high level of frequency of interaction amongst the project teams across locations facilitated transfer of specific tacit knowledge and enabled the team members to share their skills across geographic boundaries. This transfer of tacit knowledge eased the coordination in the Lake Project distributed context. This finding is in line with the literature that Jones *et al.* (1997) argue human asset specificity (e.g., culture, skills, routines, teamwork acquired through "learning-by-doing", Williamson 1985) increases the need for coordination and safeguarding exchanges amongst participants and this enhances the continuous recurring of interaction to share tacit knowledge (Williamson 1991).

Secondly, frequent exchange amongst the project teams is an important factor because it forms a condition for embeddedness, which establishes a foundation for social mechanisms to safeguard and coordinate exchanges effectively (Williamson 1985). I found strong embeddedness amongst the project teams locally and across locations. This embeddedness appears to have occurred through frequent communication and interaction amongst the teams and resulted in enhanced coordination and safeguarding of exchanges amongst the project teams, as suggested by Jones *et al.* (1997).

Thirdly, interaction frequency is considered an important element by governance literature because it is cost effective when using governance structures (Williamson 1985).

The results of the case study showed that restricted access facilitates coordination of exchanges amongst the Lake Project teams across geographic boundaries, not only through the high frequency of exchange but also through high clarity of roles. The NG literature only discussed frequent interactions as the factor that explains the impact of restricted access on coordination (Jones *et al.* 1997, Bryman *et al.* 1987, Eccles 1981, Faulkner & Anderson 1987). However, I found clarity of the project team members' roles as an additional factor that indicates restricted access impact on coordination, especially in distributed context. Because of the distribution of the Lake Project teams across geographical boundaries and the large size of the project, the roles' clarity was very critical for the teams to reduce confusion and to save time and effort in their interactions.

At each level; program, project and core team level, *"Each role in the project is clearly defined and everyone is very well aware of each roles' responsibilities and accountability" (Lake Project, IM)* based on their core skill set. This has made it easier for the project teams across locations to identify who they needed to interact with if they have a particular enquiry. In other words, having certain team members identified for certain types of exchange enhanced clarity and identification of their roles. Therefore, clearly defined roles restrict access to certain project team members for certain type of exchange depending on their exchange enquiry. This finding is consistent with the argument in the NG literature that effective application of restricted access increases identification amongst the project teams (Jones *et al.* 1997, Provan & Gassenheimer 1994), meaning having certain team members identified for certain types of exchange enhances clarity and identification of their roles.

In addition, as discussed in Chapter 5, there are strong ties amongst the Lake Project teams. Through these strong ties the teams were able to transfer and share knowledge and this resulted in enhanced identification and understanding of roles in the project. This finding is consistent with the literature that the more embedded and connected the teams, the more widely role understandings exist (Abrahmson & Fombrun 1992, Reddy & Rao 1990). In other words, the more connection and frequent interactions amongst fewer project team members, the more widely understood and identified are their roles (Abrahmson & Fombrun 1992, Reddy & Rao 1990, Granovetter 1973).

However, although the roles were clearly defined, I found many fluid roles existed in the core team. This was due to the project agile culture and cross-functional teams, but unlike other agile teams they only stepped into another role if they needed to help out. The team members' roles were moved around and overlapped with each other depending on what the day-to-day team problem was. They could just turn up in the Stand up and find out what's going on that day in 15 minutes. For example, a developer would step into tester role if there were lots of builds in queue pending testing. *"Everyone overlaps pretty well; everyone knows mostly what people are working on."* (Developer, team S)

Finally, coordination was achieved by organising the project structure around clearly defined roles at different levels responsible for certain types of exchange through more focussed, faster and more efficient interactions. This is in line with the literature argument that more connected and frequent interactions amongst participants, the more widely shared the role understanding; thus coordination is enhanced (Abrahmson & Fombrun 1991).

The coordination was achieved through allocation of additional points of contacts at each level to act as coordinator/mediator amongst the project team members. Introducing the coordinator roles enhanced coordination among the Lake Project teams by reducing confusion in interactions, work interruptions, communication overload and saving time and effort, which is critical in a large distributed agile project such as the Lake Project. Distributed agile development project teams face coordination and communication challenges due to its distribution nature but when the size gets bigger, then the coordination and communication becomes even more challenging (Prikladnicki 2007).

In summary, I found restricted access facilitates coordination of the project teams' activities across location through high interaction frequency and high clarity of roles in the project. The organisations are advised to consider to improve the frequency of the project teams' interactions and definition of the project roles in order to enhance the coordination of the teams' activities across borders.

The frequent interactions are supported by constructs such as status maximisation, where the project team members have more tendencies to frequently interact with the other team members in similar status. This is in line with the argument in literature that there are more interactions amongst fewer team members, which improves the team members' motivation and ability to coordinate smoothly (Jones *et al.* 1997).

Secondly, the frequent interactions are supported by relational contracting, where the Lake Project team members committed to values of this contract, frequently communicating across locations. This is consistent with the literature argument that when there are more focussed and frequent interactions amongst the project team members, they are able to widely share their role understanding, and therefore coordination is improved (Abrahmson & Fombrun 1991). Thirdly, strong embeddedness amongst the project teams assisted frequent exchanges amongst them (Jones *et al.* 1997).

Finally, 'ideological similarities', the new construct that merged through the case study findings, supported frequent interactions amongst the teams across geographical boundaries, where the team members with similar professional interests and skills interacted more frequently. In addition, clearly defined roles responsible for certain types of exchange facilitates coordination through more focussed, faster and efficient interactions.

6.3.3 Restricted Access Impact on Safeguarding

The results of the case study indicate that restricted access facilitates safeguarding of exchanges amongst the Lake Project teams across geographic boundaries through relational contracting and frequency of interactions (Jones *et al.* 1997, Sagers 2008), restricted access to certain team members for certain types of exchange and customers (Jones *et al.* 1997, Sagers 2008) and a centralised point of contacts. This finding is similar to the findings of Jones *et al.* (1997) and Sagers (2008).

The findings of the case study also indicates that relational contracting facilitates safeguarding by setting the expected values, objectives and behaviours of the team members across geographic boundaries for duration of the project. The findings show that the existence of strong ties amongst the teams enhanced the level of commitment and trust amongst the team members in regards to relational contracting. This is broadly keeping with what Jones *et al.* (1997) found in the literature.

Although the relational contract does not legally bind the project team members, I found that the team members (whilst giving this flexibility) did not take advantage of it and did not behave opportunistically. Instead, they feel they have a social commitment to each other through this contract, because the Lake Project team members found the relational contract was based on norms of long term commitment and cooperation (duration of the project). A business analyst indicates, *"It's about the team and what the team goals are and getting aligned with that. So usually when it comes to opportunistic behaviour, it's about not being aligned with the goals of the team"* (BA, team C). When team members found they were going to interact frequently with the other team members during the project life cycle with a set of values and expectations, it is was reasonable to follow the contract. This decreases the potential for opportunism in exchanges (Jones *et al.* 1997).

This finding is consistent with the argument in the NG literature that when the team members expect to interact repeatedly for the foreseeable future, it is logical to cooperate and this decreases the potential for opportunism in exchange interactions (Jones *et al.* 1997). Bradach and Eccles (1989) also state that the team members involved in a relational contract try to work with each other to ensure that all the team members' needs are met and everyone can benefit in the long run. In addition, the existence of strong ties amongst the Lake Project teams across geographic boundaries suggests that the relational contract involves a durable level of commitment and trust that characterises relational contracting (Jones *et al.* 1997). The Lake Project teams developed their relational contract during the project with commitment and agreement of all the teams. That was another reason to encourage the teams to behave less opportunistic. In this regard, it could be argued that relational contracting developed over a period of time may reduce transaction costs protected from opportunism (Bolton *et al.* 1994).

The results of the study indicate that there is a high frequency of interactions amongst the project teams where the team members transfer in-depth knowledge and information across locations on a regular basis. This is consistent with the argument in the literature that relational contracting is a preferred arrangement for continuous and frequent interactions to transfer and exchange of information and technology are the central aspects of the relationship (Williamson 1985, Teece 1981 & 1988).

The continued exchange (interaction frequency) amongst the Lake Project teams across locations appears to have motivated team members to build trusting relationships and better develop mutual interests and stronger ties, which are critical for a safeguard exchange in a distributed agile project such as the Lake Project. This finding is consistent with the NG literature that argues fewer team members that interact more frequently provide conditions for development of strong ties and embeddedness amongst the participants (Granovetter 1973). Jones *et al.* (1997) also notes that the frequency of interactions is a central concept of embeddedness amongst the teams. When this occurs, the project team members involved tend to see their interests and needs as aligned rather than in opposition (Granovetter 1992, Provan & Gassenheimer 1994), which reduces the incentives for opportunism across locations.

Therefore, relational contract reduces opportunisms amongst the project teams through existence of frequent interactions. This is in line with the literature where Williamson (1975) argues that frequent interactions and contacts amongst the team members across locations supports some minimum level of respectfulness and good manners amongst the team members. When there are expectations of repeated contacts and interactions among the participants, this discourages attempts to look for a small advantage in any particular interactions because the teams' hostility is controlled by the likelihood of rejection and cold-shouldering amongst the other team members.

In addition, Jones *et al.* (1997) states that frequent interactions change the project team members' attitudes towards interactions and when fewer team members are involved, it decreases the total amount of monitoring required for their interactions. Therefore, both transaction costs and the likelihood of becoming a victim of opportunistic behaviour are reduced.

Also, restrict access to certain team members for certain types of exchange facilitates safeguarding. For example, the existence of specific forums for certain project roles that is only accessible by team members of that forum based on their role in the project. This allows stronger ties to be formed amongst the teams, which results in increasing commitment and better identification with the development team and its goals (Granovetter 1973). When commitment is increased, participants in exchanges can be certain that the exchange is safeguarded against uncertain behaviours and opportunism is reduced.

Another example is where restricted access facilitates safeguarding exchanges through limiting interactions between the core team members and the customer by ensuring that only the business representatives interact with the customer. As the Lake Project IM explains*, "If core team members need to interact with the customer, they will ask the product owner. The product owner and e-commerce are real representatives from the customer for us."* Consequently, the project core team, such as developers and testers, never interact with the client directly. When a smaller number of members on the development team are allowed to interact more frequently with the customer, this leads to a reduction in the amount of monitoring required (Jones *et al.* 1997, Sagers 2004).

Finally, having a centralised point of contacts in the project restricts access to fewer project team members involved in interactions leading to less monitoring, less transaction cost and lower danger of victimisation. This point is captured by BA in team S explaining *"Centralised points of contacts do save time and monitoring effort because otherwise you would have high numbers of project team members involved in each interaction across locations."*

In summary, restricted access facilitates safeguarding of exchanges across locations through, firstly relational contracting where the team members have a tendency to work in line with each other's benefits and interests (instead of in opposition) by following and respecting the goals and values outlined in the contract. Secondly, safeguarding is facilitated through frequency of interactions amongst fewer team members. This is supported by strong embeddedness, thus developing trust, courtesy and reciprocity and expectation to continue these interactions in future, which leads to less motivation for behavioural uncertainty. Thirdly, ensuring only specific team members are involved in certain types of exchange to reduce the likelihood of opportunistic behaviours. Finally, centralised point of contacts in the project, safeguard exchanges by decreasing the amount of monitoring that is required and increasing the interaction of individuals within the team (Jones *et al.* 1997).

Overall, restricted access is applied in the Lake Project distributed agile through relational contracting, status maximisation, high level of embeddedness and the new construct that has emerged through the findings, 'ideological similarities'. Restricted access facilitates coordination of the project teams' activities (Jones *et al.* 1977, Feller *et al.* 2008, Sagers 2004) and safeguards their exchanges across locations (Jones *et al.* 1977, Sagers 2004) in line with literature, as expected.

6.4 Macroculture

As discussed in Chapter 5, the application of macroculture in the Lake Project is identified through different constructs such as shared norms, shared values, shared assumptions (Abrahmson & Fombrun 1992 & 1994, Gordon 1991, Phillips 1994, Jones *et al.* 1997) and a shared set of goals (Botezat *et al.* 2013). The following section discussed the concept of macroculture and the way it has impacted coordination of the project teams' activities and safeguarding their exchanges across locations.

6.4.1 Macroculture Concept

Macroculture is critical for understanding NG because completion of complex products and services require shared processes and structures for having effective exchange amongst the project teams (Jones *et al.* 1997). Macroculture facilitates efficient exchange amongst the project teams because when macroculture is established and maintained, the basic rules amongst the teams in all locations do not have to be recreated for each interaction (Faulkner 1987). The results of the case study demonstrated that restricted access, collective sanctions and reputation facilitate safeguarding exchanges as predicted by Jones *et al.* (1997). I also found that the social mechanisms that facilitate safeguarding exchanges also include macroculture as proposed by Feller *et al.* (2008).

In this study, the Lake Project teams shared common norms across locations by sharing agile and non-agile practices, common working standards, common strategies, approaches and language. Some of these norms are part of agile development practices (i.e. daily Stand ups, code review, pair programming), some other norms are developed through frequent interactions amongst the teams like common language (i.e. Sync), approaches and terminologies (i.e. avoiding the blame game, voicing out issues) and the remaining norms (i.e. peer review, forums) were set by the project team members themselves as parts of their working standards.

The Lake Project teams share common values and these are part of the Ocean Group values and are very similar to the value of the River, which includes honesty, courage, fairness, respect, caring and trust. The study results revealed that trust was one of the most important and critical shared values amongst the teams locally and across borders. Trust grew stronger amongst the teams across locations gradually while they practiced it as a value. For example, at the beginning of the Lake Project, team C developers were working differently than team S but after a couple of months of frequent interactions with team S, both teams learned to trust each other more and both teams developed similar ways of working and trusted the other team's standards. Some of the other shared values are part of agile values and others are sourced from their social contract values. The Lake Project teams also shared values through their socialisation activities that enabled them to create a sense of mutual interest.

The Lake Project teams shared many understandings and assumptions of the way all teams work across all locations. For example, there was a shared assumption across locations that the project team members trust that the BAs, the business and the IM will be communicating across locations in order to communicate the same message to all developers and testers. BA team C explains, *"Once we became more fluent we found it easier to communicate. Then, we started to share a common assumption that we will get the information from the BAs and they let us know which one has the highest priority etc."*

There were four major goals that were shared amongst the Lake Project teams across all locations at all the project levels, being: 1) delivering the project on time, 2) making the distributed agile component work, 3) delivering to the project technical strategy which is continuously deploying into production and 4) delivering the software product to the satisfaction of the business.

I found shared norms, values, assumptions and goals amongst the Lake Project teams across geographic boundaries. Therefore, the results of the case study indicated that a strong macroculture exists in the Lake Project and this is due to strong embeddedness amongst the teams. This finding is consistent with literature where it was found that the more connection and frequent interactions amongst the teams, the more widely shared are their norms, values, assumptions (Jones *et al.* 1997, Abrahmson & Fombrun 1992, Reddy & Rao 1990) and goals (Gerrit *et al.* 2010).

In addition, the embeddedness distributes information throughout the project network, it facilitates the development of macroculture because it enables the project members to share their perceptions and understandings across geographic boundaries (Pfeffer & Leblebici 1973) and regulates the shared norms, values, assumptions and goals through this interaction (DiMaggio & Powell 1983).

The case study results indicated that a high level of embeddedness exists amongst the project teams across locations. The embeddedness in the Ocean Group and the strong ties amongst all the project teams across geographic boundaries allows the inter-firm movement of projects' team members. Furthermore, as described in Chapter 4, the Ocean Group source the project team members from different domains and these team members are back to their original domain that they came from at the end of the project. Therefore, the team members moved frequently, they were re-assigned to different teams and back to their original teams within the organisation and this movement spreads norms, values, expectations and information about the other team members across locations (Friedkin 1982, Granovetter 1973 & 1982, DiMaggio & Powell 1983, Pfeffer & Leblebici 1973).

Another reason for such strong macroculture in the Lake Project appears to be the co-location of distributed teams throughout the project life cycle. Co-locating (introductory workshops) the Lake Project teams at the beginning of the project (at training camps) was critical given its distributed nature, as it gave the team members the opportunity to socialise, during which they shared their perceptions and understandings with each other face to face. In this regard, Gulati *et al.* (1994) explain that, the high failure rate of newly formed relationships and connections reveals how important established social processes are to support participants' interactions. Indeed, socialisation is referred to as one of the means that macroculture can be distributed and maintained (Kaufman 1960, Van Maanen & Barley 1984). The Lake Project repeatedly co-located teams from different locations throughout the project, to maintain the shared norms, values, assumptions and goals created through co-locating socialisation. For example, the teams from different locations were gathered at one location and they went through training, context briefing and socialisation meet ups.

In general, macroculture is enhanced by close geographic proximity because it increases the likelihood and ease of interaction as claimed by Jones *et al.* (1997). However the empirical results of the case study indicate that although the Lake Project teams were distributed across far geographical distances, they were able to develop strong macroculture by using the right practices such as co-locating and encouraging frequent interactions amongst the Lake Project teams across all locations. Despite technological change since the late 1990's, it is still essential to have face to face interactions in order to have effective communication, especially in distributed works (McLoughlin *et al.* 2000, Hinds & Kiesler 2002).

6.4.2 Macroculture Impact on Coordination

The empirical results of the case study show that a strong macroculture exists in the Lake Project team, both locally and across locations, which has encouraged a similar way of working in different locations. Indeed, following agile culture in the project by all the teams across locations enhanced the similar way the project teams work on different sites, which eased the coordination of exchange amongst all the project teams. This is achieved through developing a lot of common functionalities, tacit rules (Camerer & Vepsalainen 1988) and common language (Jones *et al.* 1997, Williamson 1975 & 1985) and approaches (Becker 1982, Camerer & Vepsalainen 1988) that are shared between the Lake Project teams across locations. The macroculture of the project teams contributes to the coordination and governance of the teams (Granovetter 1995). The empirical results of the case study showed that macroculture enhances coordination amongst the Lake Project teams in six different ways:

1. By unifying team members' expectations across geographic boundaries: Through socialisation, context sharing, shared goals, shared norms and increased visibility of activities so the team members do not work at cross-purposes on different sites. This finding is consistent with that argued in the NG literature where macroculture is seen to enhance coordination of exchanges amongst participants by creating "convergence of expectations" through socialisation so they do not work at "cross-purposes" (Williamson 1991:278).

In the Lake Project, the team members socialise across locations through the daily Stand ups, Retrospectives, Showcase meetings and the forums. The co-locating of the teams also assists with socialising. These socialisation practices were a great opportunity for all team members to get the same level of understanding of context and assists in aligning expectations. This encouraged similar ways of working across geographic boundaries and reduced the likelihood of the team members working at cross-purposes across locations. This eased the coordination of exchanges amongst the teams across borders. In this regard, Feller *et al.* (2008) state that, coordination issues can be resolved by having same way of working together, sharing culture and communication methods in the project.

The study shows that some shared norms amongst the Lake Project teams, such as a buddy role eased coordination across borders. This norm is mostly used when there is a new team member to the team and the Lake Project team members want to fit the new team member to the project team. The new team members start working on a task while they have a buddy, who is there to provide support to a new team member. The buddy guides and checks on the new team member on daily basis, thus making sure the new team member has an understanding of the project culture and knows how to behave, which reduces coordination challenges. In this regard, Feller *et al.* (2004) argued that macroculture enhances coordination of exchanges by ensuring new team members fit into the network's culture. The project team members do not work at cross-purposes and this was evident in the current study.

In addition, shared common goals across locations lessened the possibility of working at cross-purposes. These shared goals amongst all the project teams provided closer relationships, mutual adjustment and cultural alignment amongst the teams across locations that encouraged convergence of expectations amongst the teams across geographic boundaries. Indeed, the project teams share information to achieve common goals by adjusting their expectations to the other team members' expectations through frequent exchange of plenty of information (Shen *et al.* 2003). Therefore, the mutual adjustment amongst the project teams enhanced coordination of their activities across locations. This is in line with the argument in the literature that that sharing information and the project goals is one of the key elements of mutual adjustment (Hall *et al.* 2005) and mutual adjustment is considered as one of the coordination mechanisms in information system development projects (Rivard & Aubert 2008).

The results of the case study indicate that macroculture enhances coordination of exchanges in the Lake Project through harmonising the teams' expectations by increasing visibility and the project transparency of information and activities of the teams' across geographic boundaries. This is captured by the Lake Project team leader who says, *"Team members in China, Sydney and Melbourne have transparent visibility of each other's activities and this put them all in one page* *working in harmony. It’s critical for us as a distributed agile project to reduce a lot of challenges.”* In order to do so, the Lake Project uses a shared agile electronic ‘wall’ called JIRA, (JIRA is a tool that is used to provide visual representation of tasks and their status), to increase information visibility in order to align expectations and activities of the project teams in the distributed agile development environment. The JIRA software allows for activities and information visibility that makes it much easier for the teams at different sites to work ‘on the same page’ and not at cross-purposes, thus enhancing the coordination amongst the teams across locations. In this regard, Feller *et al.* (2008) argue that in social governance, coordination cost is reduced through increased visibility.

Information visibility means that the Lake Project teams have accurate, up-to-date information of the critical activities and processes of the other project teams across locations. This facilitates a harmony in the work environment thus lessening the likelihood of cross-purpose working, which eases the coordination amongst the project teams across locations. The enhanced information visibility enables the project teams to integrate value-adding activities as well as supporting joint decision making, as information exchange precedes the movement of physical resources (Ahmed *et al.* 1996). In this regard, Lee *et al.* (1997) argue that, timely, accurate, and relevant information is fundamental for enhancing coordination and can assist business process improvement and enable strategic organisational changes (Straub *et al.* 2002, Saeed *et al.* 2005). In addition, the presence of high levels of embeddedness and visibility of the project teams’ activities enables the use of social mechanisms to resolve exchange problems by coordinating and safeguarding exchanges (Jones *et al.* 1997, 1998).

2. By sharing a common language amongst project team members across geographic boundaries: To summarise complex information, the project teams shared symbolism and language that articulated the network macroculture (Botezat *et al.* 2013). The common language and terminologies developed throughout the project lifecycle simplified the communication of complex data and information. This facilitated sharing and transferring of complicated information, which helped to explain and transfer information much easier and faster rather than explain it over and over again. Thus, it made communication amongst the teams smoother, faster and simpler, which enhanced the coordination across geographic boundaries.

For example, there are words that the project team members used that are context specific such as ‘sync’ and agile methodology terms such as blockers, builds, stories, features, and release plan are quite common in agile methodology. As the Lake Project IM states, *“Because we know and share a context, [we] form a language around that context”.* This finding is consistent with the NG literature, in which Jones *et al.* (1997) and Williamson (1975 & 1985) argue that macroculture enhances coordination of exchanges by allowing for idiosyncratic language to summarise complex routines and information.

3. By sharing common approaches to a particular situation: Throughout the project, the Lake Project teams developed and shared tacit rules to act appropriately in particular situations. Common approaches to certain problems made it easier to deal with the problem, therefore eased the communication and coordination amongst the Lake Project teams across locations. For example, as the Lake Project IM explains, *“When we review our stories and find that there are too many points and there is more work to deliver by the delivery date, everyone knows what happens. You go to planning and you get the business into the room and work out an approach to it.”*

This finding is consistent with Camerer & Vepsalainen (1988:115) who claim that macroculture facilitates coordination amongst participants by specifying, "Broad tacitly understood rules...for appropriate actions under unspecified contingencies."

4. By sharing common agile practices: Sharing common agile practices such as pair programming and daily Stand ups encouraged similar ways of working across all locations amongst the project teams. This became possible by providing tools such as Skype and JIRA, so the teams’ work and activities would be consistent and aligned across borders, thus, making coordination amongst the teams easier. For example, as the BA from team S explains, *“We do have that front end developer in Melbourne and he is relying on pair programming because there is no one else to help him out, especially when it comes to context related business logic rules he’s got to rely heavily on the distributed team.”*

This finding is consistent with Feller *et al*. (2008) argument in the NG literature that macroculture enhances coordination amongst the teams by fostering a culture of network agility amongst them, because it provides harmony in their way of working and reduce possibility of complication in coordination.

5. *By sharing norms to prevent fear of lock-in:* In this regard, Feller *et al.* (2008) argue that preventing the fear of lock-in is an important factor in creating a sense of macroculture and it facilitates coordination amongst participants. The results of the case study indicate that the Lake Project teams prevented fear of lock-in through the two shared norms of *voice out issues* and *no blaming game*, thus enhancing coordination amongst the teams across borders.

The project teams shared the norm to voice out and make the project managers aware of any issues that had arisen as soon as possible and communicate the issue to as many stakeholders as possible before it was too late. They believed not only, *"Bad news doesn't get better with age"* (Team leader, Lake Project) but, *"Sitting on it slows everything down"* (IM, Lake Project). The voice out norm was encouraged by another shared norm called no blame game. No blaming game norm made it much easier for the team members to be able to speak out about problems and failures without having the fear of being blamed. This was very important because it helped in reducing the team members' fear of being blamed or criticised. When the issues are raised early without fear of being blamed, then the issue can be taken care of much easier and faster before it complicates coordination amongst the teams. Such an environment encourages coordination amongst the teams across geographic boundaries.

6. *By encouraging new members to be compatible with project culture:* It is also argued in the NG literature that macroculture facilitates coordinating exchanges by ensuring new team members fit into the network's culture (Feller *et al.* 2008). This was encouraged in the Lake Project through shared norms such as shadowers and buddy roles amongst the teams across geographic boundaries. Shadowers were backups for main roles in the project in case a team member was ill, on leave or needed to be changed and thus they did not have to add new team members to the project. This prevented complications and challenges by ensuring understanding of the project context and the project culture. Alternatively, a buddy role was used when there was a new team member added to the project team and they were required to have a good understanding of the project culture and context. The new team member started work on a task with a buddy (a senior team member) who provided support and briefed the new team member about the project context and culture. This made coordination of new project team members much easier and faster with rest of the team across locations.

In addition to the above discussion, Feller *et al.* (2008) state that, "Macroculture facilitates coordinating of exchanges by creating a sense of mutual interest." The results of the case study showed that the Lake Project teams were able to create a sense of mutual interest between the team members across locations through socialisation activities, such as co-locating distributed teams at various times throughout the project life cycle. During this process, the team members' discussions were more cultural and experience based, which opened the door for the members to have a better understanding of each other's interests. This understanding and awareness eased the coordination amongst the teams across borders.

An interesting point in the case study is that although the above points improved coordination of the Lake Project team across locations, they were insufficient to facilitate coordination of the distributed agile project team in all aspects. Distributed agile development projects require more indicators to use for macroculture to improve coordination. This could be due to the wide context of macroculture and the distribution concept of the project that makes coordination challenging across geographic boundaries. For example, although common agile practices aligned the Lake Project teams' activities across geographic boundaries, I found that it was insufficient. At the beginning of the project, the team members found a slight misalignment due to different working cultures between China and Australia. Team C tended to work longer hours than team M and team S in Australia, even on Fridays where it is a cultural norm for Australians to work in a more relaxed manner. Team S and team M collectively worked until 4 o'clock on Friday afternoon and this was something that team C was not prepared for. They did not understand the concept of a more relaxed attitude on Friday, so there was confusion around different ways of working on different sites.

In summary, macroculture enhances coordination amongst the Lake Project teams by unifying the team members' expectations across locations, sharing common language amongst the project teams, sharing common approaches to particular situations, sharing common agile practices, sharing norms to prevent fear of lock-in and by encouraging new team members to be compatible with the project culture.

6.4.3 Macroculture Impact on Safeguarding

The study showed that macroculture facilitates safeguarding of the Lake Project teams' exchanges across locations through shared norms and shared values by creating a 'safe-to-fail' environment and reducing the amount of monitoring required to protect the teams' transactions across geographic boundaries.

1. Shared norms:

Continuous and constant shoulder checking (peer review) between the project teams across geographic boundaries decreases the likelihood of detrimental opportunistic behaviour occurring. Issues such as the lack of context, ambiguity or misunderstanding are uncovered quickly, thus lowering the chance of having undesirable opportunistic behaviour. Constant shoulder checking (peer review) shared and used by the project teams locally and across locations, was used as a safeguard strategy to protect the project teams of possible opportunistic behaviour.

Shadower and buddy roles were also used as back up roles to safeguard replacement of the project team members when they go on leave or change during the project. The shadower/buddy learned the context, the team norms, values, goals and working culture. Therefore less monitoring was required and there was a lower possibility of falling victim to opportunistic behaviour.

The common shared norms, such as avoiding the 'blame game' and voice out issues, protects the project teams' interactions and exchanges locally as well as across geographic boundaries, by providing a safe-to-fail environment for the team members. When all of the project team members have a shared understanding that any time an issue has been raised, it can be improved without the team member being blamed in a confrontational way, the issues become much more transparent and easier to handle. The Lake Project IM explains that, *"Having] a kind of safe space within the team and being able to say that something is not working or you don't feel like you are getting value out of particular activity or a process and having that norm that there is no blame attached to that... you are kind of seeking how to improve the process for the team rather than find out whose fault it is. It prevents conflict in the team."*

Avoiding the blame game norm is about celebrating a failure and providing a trusting and safe environment for the team members to voice out issues as soon as possible. Due to distance challenges, it is especially critical to have them in distributed agile development project teams such as the Lake Project. Therefore, it decreases the likelihood of possible conflicts and behavioural uncertainties across borders.

High visibility is another shared norm that contributes to safeguarding the Lake Project teams' interactions across locations. The project team members increased the visibility of their activities by sharing and maintaining agile electronic wall (JIRA). Through JIRA, all the project teams have the visibility and transparency of each other's activities across borders, which is critical in distributed context. Monitoring to control the Lake Project teams' opportunism is very easy with high visibility or even unnecessary when the probability and the cost of their opportunistic behaviours being detected are sufficiently high (Dyer 1997). Therefore, increased visibility and transparency of the teams' activities decreases the amount of monitoring (cost) thus safeguarding their exchanges across locations.

In addition to the above shared norms, the case study result showed the teams share a common norm that negative behaviours, comments or attitudes are not encouraged by the team members across locations but positivity (like achievements) is motivating and discussed. This encourages a trusting environment and lessens the amount of monitoring required, thus safeguarding exchanges across borders.

2. Shared values:

The results of the case study indicated trust as one of the most important and strongly held values shared amongst the teams across geographic boundaries. As discussed in Chapter 5, one of the reasons for having such a trusting working environment in the Lake Project is that 'trust' is not only part of the project team's values but also part of the Ocean Group and the River organisational values. Indeed, the project teams' working culture was based on trust even before they were re-assigned to the Lake Project. That is why trust was very much embedded in their working culture and daily activities. Trust is a strong value amongst the teams in all locations and they, "*have been living that value*" (Developer, team S). This was clearly indicated in the example where team C followed Australian teams, based on a high level of trust across locations, to complete a considerable part of a task where otherwise the project completion would not be possible. This trusting and safe environment across geographic boundaries decreases the incentives for the project team members to become victim to opportunistic behaviours.

This finding is very much consistent with the argument in the inter-organisational governance literature where values such as trust, by their very nature, develop safeguards against taking advantage of transactions (Dwyer *et al.* 1987). Trust requires the project team members to perceive each other as trustworthy, and requires a willingness to give up opportunistic behaviours. Trust has been defined as, "an implicit or explicit pledge of relational continuity between exchange partners" (Dwyer *et al.* 1987, p. 19). Thus, the expectations of relationship continuity motivates the project team members to give up opportunistic behaviour, as their relationships develop because the team members come to rely on their relational norms (trust) as safeguards (Zaheer & Venkatraman 1995). Relational norms thus appear to both safeguard transactions and to enhance the collaboration within an exchange relationship (Wang & Wei 2007). In this regard, Feller *et al.* (2008) argue that the result of their case study acknowledges the importance of creating an environment of trust. Then, they note that macroculture facilitates safeguarding of exchanges by creating an environment of trust, which is consistent with the current study finding.

In summary, macroculture facilitates safeguarding exchanges by creating a 'safe-to-fail' environment through shared norms such as shoulder checking (peer review), shadower and buddy roles, increased visibility, avoiding the 'blame game' and voice out issues to protect the project teams' interactions and exchanges locally as well as across locations. Macroculture also safeguards the project teams' exchanges through shared values such as trust which is one of the most important and strong values shared among the teams across geographic boundaries amongst distributed agile project teams. This encourages a safe environment and lessens the amount of monitoring required, thus safeguarding exchanges across borders.

Overall, macroculture applied in the Lake Project through shared norms, shared values, shared assumptions and a shared set of goals. Macroculture facilitates coordination of the project teams' activities (Jones *et al.* 1977) and safeguards their exchanges across locations (Jones *et al.* 1977, Feller *et al.* 2008) in line with literature as expected. In addition, the case study result indicates that macroculture had wider and more complicated context than what has been discussed in previous studies. This social mechanism requires deeper understanding and analysing compared to other social mechanisms in NG. Next, collective sanctions are discussed in details.

6.5 Collective Sanctions

The empirical result of the case study revealed that the application of collective sanctions in the Lake Project is identified through constructs including actions to condemn the unacceptable behaviour and actions to reward the acceptable behaviour. The case study result showed that within the Lake Project team, collective sanctions for unacceptable behaviours across geographic boundaries were imposed on the team members who were low performers, played the blaming game, were either too aggressive or negative or critical and eventually became blockers to progress. On the other hand, the project team members were rewarded for acceptable behaviours through informal actions, such as words of encouragement, acknowledging and recognising the team members' achievements through financial bounce and sending thank you messages through the 'THANK YOU' system.

6.5.1 Collective Sanctions Concept

Collective sanctions are actions taken by the project teams to condemn the unacceptable behaviour or to reward the acceptable behaviour (Jones *et al.* 1997). The Lake Project identified acceptable behaviour parameters through its values, norms and relational contract. If an unacceptable behaviour happened that was in contrast with the team social contract and the Ocean Group values, then the team members from across geographic boundaries would raise it as a concern through social sessions. This is a joint act by the teams across locations, as they feel safe to talk about it openly.

The case study results showed that this reaction depends on the maturity of the team and how long the team members are working together. Usually it doesn't happen in the early phases of the project because the team doesn't feel comfortable to raise an unacceptable behaviour through social sessions in the first few months, but after a few months the team members start to voice these issues out. The reason is that during the project life cycle, close ties and embeddedness formed amongst the teams across locations, the misbehaviours were shared and the project team members acted jointly. This finding is consistent with the argument in the NG literature that collective sanctions are not possible without embeddedness since participants must know about unacceptable behaviour in order to act jointly to condemn the behaviour (Jones *et al.* 1997).

The case study results showed that the Lake Project used informal modes of sanctions such as verbal encouragement and conversation or rotating the assigned task. This is consistent with the argument in the literature that in social exchanges, disputes, conflicts, norms and values violations have to be resolved by mechanisms other than legal and formal modes (Macaulay 1963).

In addition, the results of the study indicated that, although the Lake Project team members condemn behaviours in contrast with their norms, values and goals across all locations, they do not believe in extreme negative sanctions and financial consequences straight away. Instead they believe in constructive actions and gradual escalation. Gradual escalation is more about whether the behaviour has changed over time, and if it doesn't then it will be reported as a performance issue and escalated further. This finding is not consistent with previous literature, where the severe sanctions in form of humiliation and rebuke (e.g. flaming) and the exclusion of individuals were imposed to manage violation of norms and values (Jones *et al.* 1997, Bergquist and Ljungberg 2001, Sagers 2004, Szczepanska *et al.* 2005, Feller *et al.* 2008). However, this strategy worked well in distributed agile development context of the Lake Project as the Lake Project IM already indicated, "*We found gradual escalation as an opportunity to clear doubts and misunderstanding before it got to severe sanctions.*"

The study findings showed the reason is that the Lake Project managers believe that while imposing negative sanctions may have a short-term effect, it does not bring long-term protection to the same extent as when the team members feel they are a part of the team. For example, the financial consequences will apply annually and it does not have the desired effect on daily activities of the team members. Another reason was they did not want to damage the strong bond and safe and trusting working environment that existed amongst the teams across geographic boundaries. This is because collective sanctions have their own limitations on how accurately they may be applied and one is often unable to differentiate intentional opportunism from genuine misunderstandings, especially with complex tasks under conditions of high uncertainty like distributed agile development projects such as the Lake Project (Jones *et al.* 1997).

The literature also argues that as the uncertainty increases, it becomes more difficult to recognise when participants have met or not met their obligations to one another (Jones *et al.* 1997). Due to these limitations and sensitivities, the Lake Project team members were hesitant to impose severe negative sanctions straight away and preferred to use a gradual escalation system to give time to the team members to better understand their intention and to protect the trusting and safe environment of the project.

Therefore, the Lake Project managers preferred using a gradual escalation system, encouraging positive and well regarded behaviours to motivate the safe and trusting environment and sense of belonging to the team amongst the team members. This cautious strategy worked well in the Lake Project. For example, to determine whether or not the low performance is a question of capability or attitude, less passionate and low performing team members were given more direct responsibility with more support to deliver the work assigned to them, rotate the assigned task (due to lack of interest) and go through coaching and more training if there was a gap in their skills set. The aim was to enhance the engagement level by creating a culture where people are comfortable and they can ideally pull up by high performers

6.5.2 Collective Sanction Impact on Coordination

I found that collective sanctions facilitate coordinating exchanges across locations. This finding is consistent with the proposed model of Feller *et al.* (2008) in NG literature. It is not however congruous with the argument of Jones *et al.* (1997, 1998) and Sagers (2004) that coordinating exchanges can be achieved only through the social mechanism's restricted access and macroculture.

The result of the study revealed that collective sanctions facilitate coordination of exchange in the Lake Project by (1) encouraging the team members to act in line with acceptable behaviours through a high level of transparency and visibility of the team members' activities; (2) motivating the team members to behave in a manner that is regarded as competent and professional (Feller *et al.* 2008) by all the project team members through shared norms including peer pressure, no blaming game, through promoting and rewarding acceptable behaviours and meta-norm (encouraging the team members, in particular, senior roles to constantly monitor and act against unacceptable behaviours of the other team members, otherwise they will face the consequences); and (3) through organising the project team members around agile values. Jones *et al.* (1997, 1998) argue that the presence of high levels embeddedness and visibility of activities enables the use of various social mechanisms to resolve exchange problems by coordinating and safeguarding exchanges within networks. Visibility and transparency of activities is very critical in distributed context to avoid conflicts and confusion and to save time and effort to coordinate exchanges amongst the teams.

The Lake Project teams are encouraged to act in line with the expected behaviour. This is practiced by increasing the visibility and vision of what the project expectations and goals are. The Lake Project team members are also clearly told that the project wants them to be aligned with that vision, otherwise they have to leave the project. The Lake Project used the software application JIRA (an agile electronic wall) and a 'voice out' practice to increase the visibility of the project teams' activities across geographic boundaries. JIRA (agile electronic wall) has the ability to see team members' activities across borders because all the team members have access to JIRA and through JIRA the teams' activities are transparent. The transparency of the project teams' activities across geographical boundaries is a very important element in coordinating their activities. This point is captured by the project manager explaining, *"Visibility and transparency of activities is very critical in the Lake Project distributed agile environment to coordinate members' interactions. It helps in avoiding conflicts and confusion, it also saves time and effort amongst team members across all locations."*

On the other hand, the Lake Project encouraged the project team members to speak up and voice out issues. These practices increase the visibility of their activities as well as building trust amongst the team members, thus easing the coordination of their exchange. In this regard, Feller *et al.* (2008) note that collective sanctions coordinate exchanges by encouraging firms to be transparent in their dealings with the other team members. The empirical result of the case study showed that the Lake Project encouraged the project team members to behave in a competent and professional way through shared norms that included peer pressure, no blaming game and meta-norm so that coordination across geographic boundaries would be enhanced.

Through peer pressure the team members informally monitor the other team members locally and across locations, which increases the potential of good behaviours and brings more discipline within the team. Similarly, the no blaming game norm encourages the team members to behave professionally where the team members jointly deny or prevent blaming and negativity across locations. Alternatively, meta-norm enforces acceptable social norms (Axelrod 1985). Meta-norm is about the senior team members of the teams constantly monitoring and acting against unacceptable behaviours otherwise they will face the consequences of neglecting. Therefore, meta-norm is an incentive for senior staff to act according to their responsibilities and feel accountable for unacceptable behaviours of the team members. Therefore, these norms encourage the team members to act competent and professional across all locations, thus easing the coordination.

In addition, the Lake Project encouraged the project team members to behave in a competent and professional way by promoting positive behaviours and recognition of acceptable behaviours amongst the teams. The motive behind these actions is to promote an example as a role model so the team members are tempted to adopt and engage those behaviours within the team more often because it is regarded as a very positive action by the team. This point was supported by team S developer saying that, *"…when you see another member is rewarded and regarded highly because of professional behaviour, you will try your best to behave similarly to gain the trust and respect of other members as well."* The reason for their motivation is that the acceptable behaviour or high performance is regarded as a very positive action by the teams across locations and gains their trust and respect.

The rewarding actions and incentives for appropriate behaviours and high performance in the project had a positive impact on coordinating the project teams across locations. I therefore conclude that collective sanctions, as well as restricted access and macroculture, facilitate coordinating exchanges. The Lake Project management organised the project team members around the project values, which included agile values, by constantly reminding the team members of the values. This encouraged a similar way of working across locations and thus enhanced coordination.

In summary, collective sanctions enhance coordination amongst the Lake Project teams by encouraging the team members to act in line with acceptable behaviours through increasing transparency and visibility of the teams' activities, by motivating the team members to behave in a manner that is regarded as competent and professional through shared norms such as peer pressure, no blaming game and meta-norm and by organising the project team members around agile values.

6.5.3 Collective Sanctions Impact on Safeguarding

The empirical result of the case study showed that collective sanctions facilitate safeguarding exchanges, as predicted and proposed by Jones *et al.* (1997), Sagers (2004) and Feller *et al.* (2008). Collective sanctions safeguard exchanges, because they define and reinforce parameters of acceptable behaviour by demonstrating the consequences of violating norms and values (Sagers 2004). Indeed, the case study findings revealed that collective sanctions facilitate safeguarding exchanges in a similar manner as they facilitate coordinating exchanges, consistent with Feller *et al.* (2008) claim that collective sanctions safeguard exchanges in the same manner they coordinate exchanges.

The case study findings indicated that the strong embeddedness and trust amongst the teams across borders facilitated the diffusion of awareness about misbehaviour or good behaviours of the project members amongst the other team members. In this regard, Jones *et al.* (1997) stated that collective sanctions are not possible without embeddedness because the team members must know about misbehaviours in order to condemn or reward perpetrators. In addition, peer pressure and meta-norm encouraged the team members to constantly monitor each other. Such knowledge about the other team members made it possible for the team members to act and condemn the misbehaviour immediately, which acts as a safeguard to protect the other teams from any unacceptable and unappropriated behaviour.

This is consistent with the argument in the literature that in social exchanges, one approach for social sanctions primarily involves, "the mutual monitoring between the participants and the rapid dissemination of information about the capability of participating companies" (Hagen & Choe 1998:595). In addition to resolving conflicts, this process also helps avoid conflicts as being aware of the effect on their reputation; the project team members will act more prudently regarding expected cooperation. Thus, collective sanctions also serve as a disincentive mechanism against opportunism.

The case study result indicates that both actions to condemn and reward the behaviours reduces behavioural uncertainty by increasing the cost of opportunism although gradually, decreasing the cost of monitoring, and providing incentives to monitor the other team members for unacceptable behaviours. This finding is consistent with argument in the NG literature (Jones *et al.* 1997, Hagen & Choe 1998, Feller *et al.* 2008).

High visibility of activities encouraged the team members to behave in line with acceptable behaviours thus less monitoring is required to safeguard. Voice out issues is a practice used by the Lake Project team members to increase the visibility of the project teams' activities across geographic boundaries. The intention behind this practice was to promote a 'safe-to-fail' environment for all the team members across geographic boundaries, thus they feel safe to raise the behaviours that are in contrast with the project team norms, values and goals. This is captured by the Lake Project IM that explained, *"We have a guy in China say 'Hey guys I've done this and I really screwed up and it's not going to work and I'm sorry' and everyone goes, 'Well let's correct it and let's do not do that again.' So we say let's celebrate failure and make sure that they are not locked in and isolated."*

The safe-to-fail environment reduced the amount of monitoring required in all locations and increased trust amongst the team members, thus protected teams from possible opportunistic behaviours. The results of the case study also revealed that collective sanctions motivate the team members to behave in a manner that is regarded as competent and professional through shared norms. The reason for this motivation was to encourage the other project team members across locations to use the rewarded behaviour as their role model and get motivated to behave the same in order to gain positive reputation and gain trust of the other team members. I found that encouraging such practices across the project team safeguarded the teams from possible inappropriate and opportunistic behaviour.

The Lake Project teams jointly deny and condemn blaming game and negativity across locations, thus reducing opportunism. Peer pressure was another norm used by the Lake Project teams as a safeguard against wrong behaviours, where the team members informally monitor the other team members locally and across locations. This monitoring increases the potential of good behaviours and brings more discipline within the team. Similarly, meta-norm decreased the amount of monitoring effort. The scope of monitoring extended the project team level where all the team members had the responsibility of observing the conduct of all the other team members. Observations were quickly shared with the aim of influencing the general behaviour of the team member.

The case study findings showed that meta-norm and peer pressure encourage the team members to act in a competent and professional manner across all locations and hence reduce the possibility of the teams falling victim to opportunistic behaviours. This was supported by the Lake Project IM stating, *"We questioned why we did not take care of the situation, why we did not monitor the case and did not report it on time."* The Lake Project team leader also added, *"As a lead you should be looking at monitoring inappropriate behaviours and you should be looking at having a more positive attitude."* This finding is consistent with the argument in the NG literature that collective sanctions safeguard exchanges by decreasing monitoring costs and providing incentives to monitor partners (Jones *et al.* 1997).

In addition, the study result indicated that the gradual escalation system helped to maintain the trusting environment amongst the project team members across locations and protected their interactions. This is because it had the benefit of providing a second chance for wayward team members to correct their behaviour, which safeguarded the teams from any blaming games and unnecessary sanctions if there had been a misunderstanding about their behaviour. Indeed, it reduced opportunistic behaviour by providing greater incentives for cooperation.

In summary, collective sanctions facilitate safeguarding exchanges in a similar manner. They coordinate the project team activities across borders meaning by encouraging the team members to act in line with acceptable behaviours through high levels of transparency and visibility of the teams' activities, by encouraging the team members to behave in a manner that is regarded as competent and professional across borders through shared norms such as peer pressure and no blaming game and rewarding acceptable behaviours and finally by organising the project team members around agile values. They all encourage a safe environment and to lessen the amount of monitoring required, thus safeguarding exchanges across locations.

Overall, collective sanctions were applied in the Lake Project distributed agile through actions to condemn the unacceptable behaviour and actions to reward the acceptable behaviour. The case study findings are in line with Feller *et al.* (2008) in NG literature that collective sanctions facilitate coordination of the project teams'. However, it is not congruous with the argument of Jones *et al.* (1997, 1998) and Sagers (2004) that coordinating exchanges can be achieved only through two social mechanisms of restricted access and macroculture. In addition, collective sanctions safeguarding the teams' exchanges across locations (Jones *et al.* 1997, Sagers 2004 and Feller *et al.* 2008) are in line with literature as predicted. Next, reputation is discussed in detail.

6.6 Reputation

The application of reputation in the Lake Project network is identified through constructs including social expectations (Feller *et al.* 2008) and individual expectations. Social expectations include the extent to which the project team members' expectations are important to the other team members and individual expectations are the extent to which the individual project team members' expectations are important to themselves.

6.6.1 Reputation Concept

Reputation is identified as the perception that project team members hold about another team member's character, knowledge and skills across the project teams. It was anticipated that an individual's reputation would be particularly important in the Lake Project context because the project was characterised by uncertainty and customisation (Jones *et al.* 1997). Jones *et al.* (1997) also argue that reputation is important in NG because it relays information about prior behaviours and serves to deter opportunism and guile (Jones *et al.* 1997). In addition, reputation was a mechanism to understand the other team members' perceptions of character, capabilities and intentions of the other team members (Bergquist & Ljungberg 2001, Gallivan 2001, Hars & Ou 2002, Sagers 2004, Stewart 2005, Szczepanska *et al.* 2005). It also has been shown reputation plays a role in selection of entities within a network (Gemser & Wijnberg 2001).

The case study result revealed that it was indeed important for the Lake Project team members to have a positive reputation. I found the reason behind this was that majority of the team members were highly skilled individuals and highly regarded within the team and wider network before joining the Lake Project. This motivated them to maintain a positive image as would be expected from someone. This finding is consistent with the argument in the literature where models of reputation presume a tight coupling between individuals' past activities and future expectations (Weigelt & Camerer 1988). Other motives were increased trust of the other team members and a high level of visibility amongst all of the project team members.

In addition, the Lake Project was a high profile project within the organisation, being the first distributed agile project. However, although the Lake Project team members faced a lot of challenges due to the distributed nature of the project, they were able to get through challenges to the extent that they became a reference point for the other project team members. This indicates the importance of have a positive image for the Lake Project team members.

The application of reputation in the Lake Project network is identified through social expectations and individual expectations. The case study result showed that the project team members' social expectation from the other team members was important. Factors that motivated team members to achieve this include the agile iterations work base, the high visibility of the team members' activities and the importance of delivering to the satisfaction of the business, not just from the business point of view but also from the other IT teams within the larger network. The Lake Project was the first distributed agile project for the Ocean Group and it was important that the project was completed successfully. That is why fulfilling social expectations was very important for the Lake Project team members. Feller *et al.* (2008), in their case study of NG, identified social expectations as one of reputation indicators. In this regard, they note that it was important for participants to fulfil their obligations to the other team members to maintain their reputation within the network.

In addition, it was found that although achieving social expectations was important to the project team members, it had higher priority for team C because of their position in the project as an IT consulting company and team S and M were their customer. This partnering arrangement made gaining a positive reputation and fulfilling social expectations even more critical and important for the China team. Therefore, the condition and position of individuals in the project impacted their motive to maintain a positive reputation.

The case study findings also indicated that there was a strong tendency amongst all the Lake Project teams across locations to gain the other team members' trust and be considered as a competent and reliable professional. Therefore, they considered it important to fulfil their obligations towards the other team members locally and across locations in order to maintain their positive reputation. This is consistent with the NG literature where Feller *et al.* (2008) argue that one of the indicators of reputation in their case study was the participant considered it important to be regarded by other team members as being professionally competent, reliable and trustworthy.

This was very visible in the Lake Project because the project team members were constantly seeking information about the project context and background and wanted to know why they were doing things in order to help them to make better decisions. Indeed, they always wanted to try to prove themselves and make sure that they were a very valuable asset of the team, because team members with a positive reputation became more valuable and involved in the team (e.g. in discussions and decision-making activities) and were awarded with more advanced tasks. This is consistent with the argument in literature that the higher an individual's reputation is, the more valuable she/he becomes (Kilduff & Krackhardt 1994). The study results also showed that although all the team members across locations considered it important to gain a positive individual reputation, this sense was stronger amongst team C in China than the Australian based teams (team S and team M). Team S and team M were the 'customer' of team C.

6.6.2 Reputation Impact on Safeguarding

The results of the case study revealed that reputation facilitates safeguarding exchanges as predicted and proposed by Jones *et al.* (1997), Sagers (2004) and Feller *et al.* (2008). However, I found different indicators of reputation to protect the team members' exchanges across geographic boundaries from these previous studies. The case study showed that not only were the project team members' social expectations from the other team members important, the team members consider it important to fulfil their obligations towards the other team members across borders. Therefore, when reputation is perceived as important, individuals will be motivated to protect their reputations by avoiding deceptive or self-interested behaviour (Weigelt & Camerer 1988, Sagers 2004). Indeed, the teams' interactions were safeguarded because the team members cared about their reputations (Markus *et al.* 2000), hence a key finding of this study is that the underlying driver of safeguarding in the Lake Project is the desire of the project team members to protect their reputations.

The result of the case study, as set out in Chapter 5 indicated that reputation facilitates safeguarding of the Lake Project teams' interactions across locations through the existence of practices to prevent unexpected and opportunistic behaviours. These practices encouraged the project team members to maintain their positive reputation thus less monitoring was required and there was reduced motivation for behavioural uncertainty and opportunisms. Reputation enhanced safeguarding of the Lake Project teams' exchanges through the following practices:

1. High level of trust between members across locations

One of the important findings of the case study was that the teams' interactions across locations was protected due to the existence of a high level of trust in the team members' ability to perform tasks based on previous presentation of their quality of work. Although the distributed nature of the project made it challenging to do tasks across geographic boundaries, the existence of trust based transactions between the team members across borders reduced the amount of monitoring required. Thus, trust based reputation protected the teams from being victim of opportunistic behaviours. This finding is consistent with the argument in the literature that trust is an important indicator to evaluate individuals' abilities to manage their reputation and assure stability of their behaviour in the future (Lewicki & Tomlinson 2003). In addition, this trust also played an important role in establishing ethical behaviour (Crampton & Dees 1993).

2. Resourcing project with members demonstrated positive reputation and competency

Due to the sensitivity and importance of the Lake Project, majority of the Lake Project team members were assembled based on their skills and their delivery reputation in previous projects. Thus, having highly regarded the team members with a positive reputation and high social status (based on previous quality presented) was a mechanism that the Lake Project used to safeguard and protect the future interactions between the team members and reduce the occurrence of opportunistic behaviours as much as possible. This was demonstrated by the Lake Project IM who stated, *"You do notice that there is a tendency for certain people to go to a particular person for certain exchanges and enquiries. We know the background based on what the team leader advised us."* In this regard, Sagers (2004) argued that when the team members of the project perceive reputation as an important part of their identity, they are more likely to manage conflict within the project in order to preserve their identity and social status. Therefore, NG mechanisms such as reputation allow well-coordinated exchanges amongst teams to be safeguarded.

3. Detecting and diffusing team members' behaviours

Voicing out issues immediately was one of the social expectations that was used extensively by the project teams. This gave them a positive reputation that they were doing the right job otherwise they would have discussed it with the other team members and sought help and support. In addition, the case study findings indicated that having strong embeddedness amongst the Lake Project teams across locations enhanced the flow of information about the team members' reputations and their behaviours throughout the entire team locally and across borders as a safeguard. This distribution of information was done via word of mouth and frequent role rotations in the team. This facilitated the teams, as they were well informed about other team members' reputation and trustworthiness. This encouraged the Lake Project teams to prefer interactions with other team members that they knew to have the skill set and knowledge they were looking for. This point is captured by a developer in team S who stated, *"Majority of the time I prefer to interact with team members that I know [reputation wise] whatever I ask about that subject, it's probably going to be quicker, in particular when I am in a rush. I definitely think other people in the team do the same thing."*

In this regard, Jones *et al.* (1997) argue that reputation reduces behavioural uncertainty by providing information about the reliability and goodwill of other team members. Reputation safeguards against opportunism and acts as a mechanism to prevent unacceptable behaviours that could result in a negative image and reputation. This finding is consistent with the NG literature where embeddedness is seen to enable reputation to safeguard exchanges because it communicates the detection of, and serves to deter, deceptive behaviour, which in turn enhances cooperation (Parkhe 1993).

As mentioned previously, one of the key findings was that all the project team members condemned negative gossips and comments across locations because the Lake Project team members believed that it creates negativity in the project and this damages trust. As such, information about the positive behaviours and reputation was often distributed amongst the project teams. This finding is in contrast with Jones *et al.* (1997) who argue that for reputation to be effective, information about participants' behaviour (both negative and positive) must flow throughout the project team.

4. High visibility of team members' activities encouraged them to gain and maintain positive reputation

Positive or negative behavioural patterns were pointed out easily by the other project team members due to high level of visibility of activities that existed in the project. The existence of such visibility of activities across locations motivated the project team members to earn positive reputation by presenting acceptable behaviours towards the other team members as much as possible, and fulfil expectations of the other project team members. In other words, the existence of such visibility and transparency of the teams' activities made fulfilling social expectations and individual expectations very important in the project. When the team members saw their activities were transparent to the other teams, this motivated them to act as competently as possible, not only for their interests but to protect their reputation. Thus, behavioural uncertainty is reduced and the teams' transactions are protected. This is captured by the Lake Project IM that stated, *"…when team members see their activities are visible to all project members and there is nothing to hide, they do their best to complete their tasks as good as possible to gain and maintain positive reputation. Then other members can trust them more."*

The case study found that high visibility enhances reputation (Fombrun & Van Riel 2004). In this regard, Fombrun (2005) argued that visibility is one of the key dimensions of reputation and strong reputation is a result when companies earn visibility. The author further states that network relationships, "Reduce the risk of opportunism by higher visibility of transactions." Therefore, high visibility of the teams' activities enhances safeguarding of their transactions.

5. High profile and importance of the Lake Project as an incentive to protect reputation

The importance and high profile of the Lake Project was a good incentive for the team members to protect their current reputation. In addition, the project team members regarded being part of the Lake Project team as an excellent opportunity, as it could be used as a reference for their future work opportunities. This point is supported by a team S developer who explains that, *"Having a positive reputation in such a high profile project is considered very important by team members for their current and future career."* Thus, the importance of being part of such a high profile and important project, such as the Lake Project, decreased the incentives for unacceptable and opportunistic behaviour in the project. This finding is consistent with the argument in the literature that when the project team members perceive reputation to be an important part of their identity, they are more likely to protect exchanges amongst the teams (Sagers 2004).

In summary, reputation facilitates safeguarding of the Lake Project teams' exchanges across locations through the existence of a high level of trust between the team members across locations, resourcing the project with positive reputation and competent team members, detecting and diffusing the team members' behaviours, high visibility of the teams' activities and the high profile and importance of the project.

6.6.3 Reputation Impact on Coordination

One of the key findings of the case study was that reputation facilitates coordination of exchanges. However, this finding is inconsistent with previous literature, where scholars found coordination is facilitated by other social mechanisms such as restricted access (Jones *et al.* 1997, 1998, Sagers 2004, Feller *et al.* 2008), macroculture (Jones *et al.* 1997, 1998, Feller *et al.* 2008) and collective sanctions (Feller *et al.* 2008). But there was no relationship found between reputation and coordination.

There is a lot of agreement in the literature as to what impacts coordination. For example, Jones *et al.* (1997, 1998) argue that where high levels of embeddedness and visibility of activities are present, coordinating exchanges can be achieved through two social mechanisms, restricted access and macroculture. In addition, Sagers (2004) argues that coordination of exchanges is enhanced through only one social mechanism, restricted access. Furthermore, Feller *et al.* (2008) claim that there are three social mechanisms that facilitate coordination of the teams' interactions including restricted access, macroculture and collective sanctions. Results of the case study, however, suggest that all four social mechanisms, restricted access, macroculture, collective sanctions and reputation enhance coordination of exchanges amongst the project teams across borders. In the literature, Ching *et al.* (1996) proposed a reputation oriented coordination model within a network and argued that participants' reputations strongly influence coordination. They further claim that reputation is a coordination mechanism.

The results of the case study in Chapter 5 indicated that reputation facilitates coordination in the same manner as it facilitates safeguarding exchanges across locations. I found that reputation facilitates coordination of the Lake Project teams' activities across geographic boundaries through practices that encouraged the project team members to maintain their positive reputation, thus they had more tendencies to coordinate their activities. When the project team members across locations perceived social and individual expectations as an important part of their character, they were more eager to demonstrate and maintain a positive reputation. Therefore, such tendencies encouraged them to coordinate much easier with the other teams, thus enhancing coordination across borders.

1.7 High level of trust between members across locations

The results of the study showed that teams' interactions across locations was easier due to the existence of a high level of trust in team members' abilities to perform tasks based on the previous presentation of their quality of work. Building and maintaining a positive reputation throughout the project with the other project team members made the other team members trust them more in their ability to perform the tasks and this enhanced coordination amongst them. This is consistent with the argument in the literature where Stephenson (2005) states that trust clearly would have to exist to encourage inter-organisational coordination. The author further argues that trust plays a vital role in establishing the conditions for effective coordination amongst participants. Indeed, scholars examining the role of trust have widely acknowledged that 'trust can lead to cooperative behaviour amongst individuals, groups and organisations (Jones & George 1998, p. 531).

In addition, scholars agree that trust is an essential attribute for coordination (Noteboom and Six 2003). Further, Lorenzen (2001/2002) suggests that trust is a dominating coordination mechanism and firms are able to reduce coordination cost through the social trust that exists amongst the team members. I agree with the literature in this regard that, trust-based reputation enhances coordination of exchanges amongst the project teams across locations.

2.7 Resourcing the project with members with a positive reputation and demonstrated competency

The results of the case study in Chapter 5 revealed that the practice of resourcing the project with competent and professional individuals based on their particular skill set and reputation, made the team members expect to get along quicker because they had the skills and the reputation to deliver. In other words, there was an expectation that the team members expected the other team members to fulfil their responsibilities based on their reputation and this helped the team to get along much easier. It saved them time and effort and improved the coordination amongst them.

3.7 Detecting and diffusing team members' behaviours

The findings of the study showed that existing strong embeddedness amongst the Lake Project teams across locations enhanced the flow of information about the team members' behaviour. This facilitated the teams to be well informed about the other team members' reputation and trustworthiness. However, as mentioned before, the Lake Project encouraged spreading positive behaviours in order to maintain the trust and positive environment amongst the teams across borders. This also enhanced the existing ties and embeddedness amongst the teams, thus facilitating the coordination of exchanges amongst the project teams across locations. In this regard, Lorenzen (2001/2002) states that the more closely embedded a community, the more efficient coordination it facilitates. On the other hand, the result of the study indicating negative reputation complicates coordination, as the team members hesitate to interact with non-competent team members.

4.7 High visibility of team members' activities encourage them to gain and maintain positive reputation

The case study results revealed that high levels of visibility and transparency of the project teams' activities to the other team members across locations motivated them to act as competently as possible so as to gain or maintain their reputation. Thus, transparency and visibility is an enabling factor for reputation to facilitate coordination. In this regard, Fombrun and Van Riel (2004) argue that high visibility enhances reputation. The transparency of the teams' activities also encouraged their activities to be aligned with the norms, values and goals of the project. Thus, this harmony of activities across geographic boundaries enhanced coordination of their exchanges. This finding is consistent with the argument in literature that the visibility of activities in the workplace facilitates coordination (Bellotti & Rogers 1997, Newman & Landay 2000, Sellen & Harper 2001, Chong 2006).

5.7 High profile and importance of the Lake Project as an incentive to demonstrate positive reputation

The results of the current study showed that the project team members had a desire to maintain and enhance their own reputation and career due to the importance of the Lake Project within the Ocean Group and wider network. It was important for the team members to be part of the Lake Project because it was a high profile and important project not only within the Ocean Group but also in the Information Technology network. This motivated the teams to coordinate their interactions across locations.

In summary, reputation facilitates coordination in the same manner as it facilitates safeguarding exchanges across locations. This facilitation is through high levels of trust between the team members across locations, resourcing of the project with the team members who have a positive reputation and demonstrated competency, detecting and diffusing the team members' negative behaviours, high visibility of the team members' activities and the high profile and importance of the Lake Project.

Overall, reputation applied in the distributed agile Lake Project through social expectations (Feller *et al.* 2008) and individual expectations. Reputation facilitates safeguarding of the teams' exchanges across locations as proposed by Jones *et al.* (1997), Sagers (2004) and Feller *et al.* (2008) in NG literature. Reputation also facilitates coordination of the teams' activities across borders. However, this finding is not consistent with the previous literature. Jones *et al.* (1997, 1998) argue that coordinating exchanges can only be achieved through restricted access and macroculture. Also, Sagers (2004) argues that coordination of exchanges is only enhanced through restricted access. Furthermore, Feller *et al.* (2008) claim that coordination of the teams' interactions can only be achieved through restricted access, macroculture and collective sanctions. The case study found that reputation facilitates coordination of the project teams' activities through similar indicators that safeguards their exchanges across borders. However, these are different indicators of reputation that were discussed in these previous studies. Next, interactions between social mechanisms are discussed in details.

6.7 Revised Network Governance Model

Consistent with the extant research on social mechanisms, restricted access, macroculture, collective sanctions and reputation, were agreed by study participants as enabling the coordination and safeguarding of exchanges between the project members. However, the case study analysis revealed that the functioning of these mechanisms, and how they relate to coordination and safeguarding exchanges is somewhat different from the one proposed by Jones *et al.* (1997).

Previous case studies of NG illustrate that application of social mechanisms differ across contexts. Consequently the way they resolve exchange problems through these mechanisms will also be different. This could be due to the reason that social mechanisms act differently in different contexts such as online communities (Sagers 2004, Feller *et al.* 2008, Bogg 2010), supply chain (Wathne & Heide 2004), distributed agile development projects (the current case study) and film industry, although their fundamental will not change. Based on the specific requirements of these differing environments, different macroculture will be formed that have a different effect on other social mechanisms; therefore they act differently in coordinating and safeguarding exchanges. For example, in the case study the context of distribution was important to be considered. In OSS or online communities the working environment and the culture that teams work in, requires other points to be taken into consideration. Therefore different relationships amongst social mechanisms will be formed and they impact coordination and safeguarding differently.

Another interesting point is that, even when comparing Jones *et al.* (1997) model within the same context such as online communities (open source services), the proposed models by Sagers (2004), Feller *et al.* (2008) and Bogg *et al.* (2010) are not similar. For example, Jones *et al.* (1997) proposed model indicates that coordinating exchanges is enhanced by macroculture and restricted access, where Feller *et al.* (2008) in their study of open source service networks argue that coordination is facilitated by collective sanctions as well. On the other hand, Sagers (2004) claims that coordinating exchanges is facilitated by one social mechanism only, restricted access. In this regard, Bogg *et al.* (2010) opposed other studies in the literature and claim that macroculture is the only social mechanism of NG that facilitates coordinating and safeguarding exchanges.

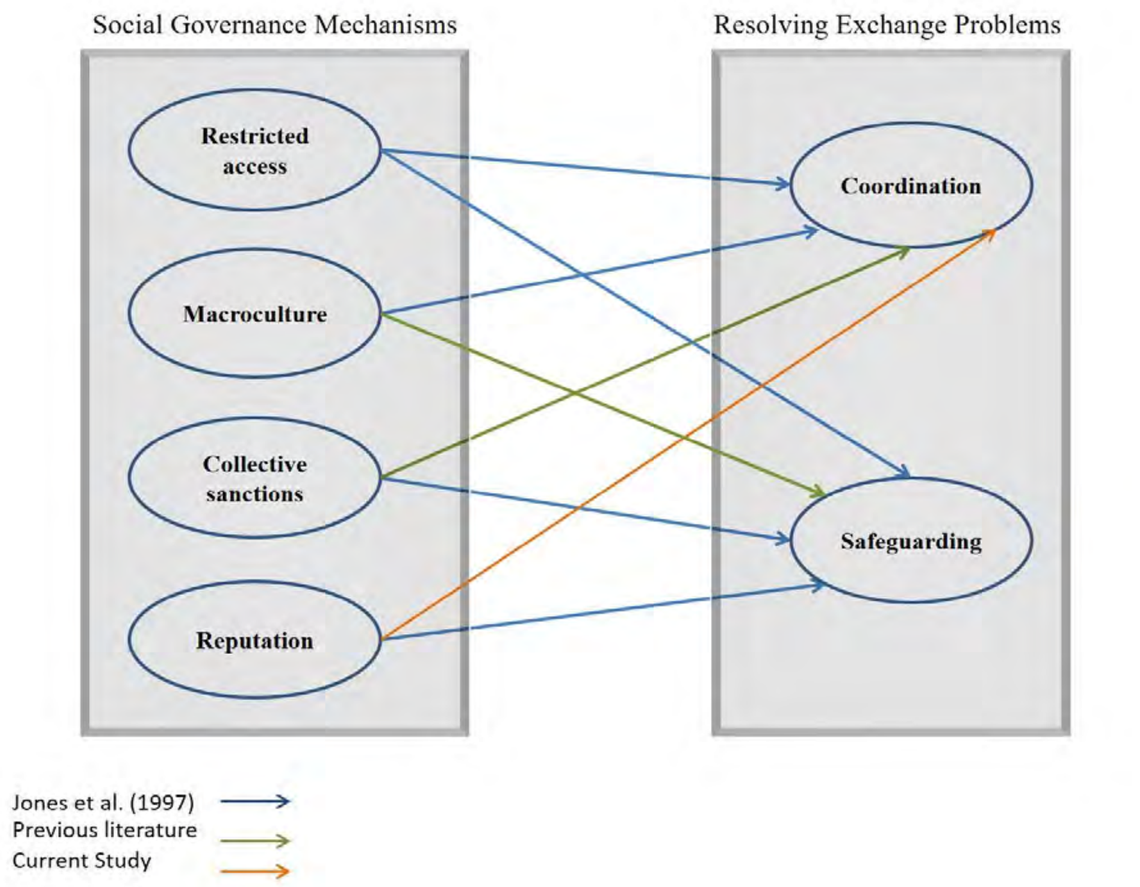
Alternatively, scholars have different arguments in the literature about how social mechanisms safeguard exchanges in NG. Safeguarding exchanges is enhanced by restricted access, collective sanctions and reputation as proposed by Jones *et al.* (1997) and Sagers (2004). However, Feller *et al.* (2008) claim that safeguarding is facilitated by macroculture, collective sanctions and reputation. On the other hand, Bogg *et al.* (2010) argue that safeguarding exchanges is enhanced only by one social mechanism, macroculture.

While previous studies indicated that coordinating and safeguarding exchanges are enhanced by some of the social mechanisms at each study, the case study result revealed that contrary to the model proposed by Jones *et al.* (1997) and previous studies of NG, safeguarding and coordinating exchanges can be facilitated by all four social mechanisms, restricted access, macroculture, collective sanctions and reputation (Figure 6.2).

Bogg *et al.* (2010) in their study of social mechanisms in online communities proposed that restricted access, reputation and collective sanctions facilitate macroculture. However, the empirical case study result does not confirm that. Although the current study result indicated that macroculture is the most important of all social mechanisms, other social mechanisms are also required to act beside macroculture in order to resolve exchange problems effectively. In addition, Bogg *et al.* (2010) proposed model indicates that the relationship of restricted access, collective sanctions and reputation with macroculture is one sided, and it does not show any relationship or interaction from macroculture to the other three social mechanisms. But the study result showed that interaction between all social mechanisms is double sided. Furthermore, the study indicates that macroculture has the most influence on the other social mechanisms.

In conclusion, the study confirmed all of the previous lines on the model. The current research study has shown the Jones *et al.* (1997) NG model basically works, however the model was incomplete and some new relationships emerged. Based on what the results are telling us we recast the model as below.

Figure 6. 2 Revised Model of Adoptive Governance in Distributed Agile Development Projects



Adapted from Jones *et al.* (1997)

6.8 Other Noteworthy points

The above section sets out the refine research model and discusses the formal results found in answering the research questions. The purpose of this section is to look more broadly at the practices and useful points that was noted while undertaking this research. This section outlines a series of points noted from the current research providing indications that can be useful for both businesses and scholars. For example, the potential interactions between social mechanisms and the importance of macroculture, noteworthy business practices to improve coordination and safeguarding in distributed agile development projects. These points are explained in more detail in the following sections. The current study is much grounded in practice and it would be negligent of me not to discuss these points that provide good guidelines for businesses.

6.8.1 Potential Interactions between Social Mechanisms

It was not the intent of this thesis to investigate the interactions between social mechanisms however in conducting the research the researcher noted some interesting points that might be useful for both businesses and researchers. The following section explains my view of the potential interactions between the social mechanisms but it needs further research as noted and suggested by Jones et al. (1997).

While conducting the current study, I noted that the four social mechanisms of NG (macroculture, restricted access, collective sanctions and reputation) are not independent from each other but indeed they interact. . In this regard, Jones *et al.* (1997) suggested that an empirical study was required to examine the interaction of the social mechanisms of NG and their influence on coordinating and safeguarding exchanges, because this could provide important insights into social mechanisms. They argue that social mechanisms are critical to effective functionality of networks; therefore there is a need to have a better understanding of how social mechanisms are interacting with one another. The interaction of these social mechanisms in NG enhances cooperative behaviour required for customised, complex tasks under conditions of uncertainty (Schroeder 1995). In addition, Jones *et al*. (1997) state that there is a need in literature to understand whether some social mechanisms are more important than others. Jones *et al.* (1997) recognise the potential for interdependencies between the social mechanisms, while Feller *et al.* (2008) note a more expansive role for macroculture in open source software development groups than that originally envisaged by Jones *et al.* (1997).

The following discusses how social mechanisms interact as per the case study. I also argue that macroculture is the widest and most important social mechanism of NG. However, one needs to keep in mind that the Lake Project was an important and high profile project and resourced from well-regarded and highly skilled team members. The Lake Project was the first distributed agile project for the Ocean Group and simply there was so much effort put into it because it could not fail. The same result might not have been achieved if the project was not high profile.

Macroculture and Collective Sanctions

One of the points that was noted while conducting the study was that macroculture and collective sanctions interact with each other. There are some elements of macroculture, such as shared norms (i.e. peer pressure, no blaming game, voice out issues) and values (i.e. agile values, trust), which become part of collective sanctions over time as well as some collective sanctions becoming part of macroculture as well.

For example, in this study, some elements of macroculture like avoiding the blame game and discouraging negativity become part of the collective sanctions and vice versa. Discouraging negative attitudes and comments and no blaming game was a practice that was used by the project teams during the project lifecycle to condemn the unacceptable behaviour of blaming others. However, the teams used them continuously and repeatedly so that they become shared norms and part of the macroculture in the Lake Project. You can expect explicitly applied collective sanctions to become shared norms and part of macroculture over time. Indeed, collective sanctions and macroculture cross over each other. Even the definition of collective sanctions (punishment by the team when team members breached the project macroculture; norms, values and goals) indicates the relationship between these two social mechanisms.

Collective sanctions are assessed based on how participants behave according to the macroculture. Unacceptable behaviours are those project team members' behaviours that violate the project norms, values and goals. Relatively acceptable behaviours are those behaviours that are in accordance with the project norms, values and goals. In other words, the Lake Project team members identify the acceptable behaviour parameters (collective sanctions) through their shared values and norms (macroculture). This is how macroculture and collective sanctions overlap. In addition, the empirical results of the case study showed that the Lake Project managers encouraged the team members to behave in a competent and professional manner through various shared norms, such as peer pressure, no blaming game and meta-norm in order to enhance coordination across geographic boundaries.

Over time collective sanctions become normal and go from explicitly applied collective sanctions to shared norms of macroculture. In this regard, Sagers (2004) argues that collective sanctions have the effect of increasing the incentive to follow norms. However, the current study points out an interesting point, that unless these collective sanctions are in harmony with the existing macroculture, they will not last long and hence do not become part of macroculture (norms).

When these sanctions do not become norms, they have to be enforced continuously, which increases monitoring costs and makes safeguarding exchanges more challenging. Therefore, in order to have effective collective sanctions, organisations need to use practices that are compatible with their macroculture in order for them to become normalised. If sanctions do not become norms, they have to be enforced continuously, which increases monitoring cost and will not have a positive effect. On the other hand, when sanctions become part of the culture, they are automatic and become the way the team works.

For example, in the Lake Project, they do not believe in imposing severe negative sanctions, instead they encourage positive attitudes by rewarding the right and acceptable behaviour across locations. The reason is clear; the Lake Project wants to maintain the compatibility of collective sanction practices with their macroculture. The Lake Project culture encourages positivity and a collaborative, safe and trusting environment through using different practices across geographic boundaries. They reward positive attitudes to become role models for others to map the same behaviours. Introducing appropriate sanctions to the project team normalised them and they become part of the Lake Project culture as shared norms between the teams across geographic boundaries.

The reason for using compatible practices is because once collective sanctions become part of the culture they are automatic and become the way the team works, thus lowering monitoring costs and the coordination effort required. In other words when the right macroculture is applied and shared amongst the team members, there is less need for imposing negative sanctions and a minimal monitoring regime is required, thus making it easier to coordinate and safeguard exchanges.

This finding is consistent with the argument in the literature that a cooperative macroculture facilitates collective sanctions. First, if the project teams develop a group culture in which collective interests are values, they will be more comfortable with mutual exchange over time (Ekeh 1974). Second, a strong macroculture ensures that collective sanctions, whenever needed, will be imposed collectively. The power of social sanctions comes partially from reinforced messages – that is, various parties delivering the same information about a team member's norm violation. Thus, sanctions will be more effective if the project team members share norms and know what message to spread when norms are violated (Das & Teng 2002).

This is consistent with what happened in this study. The Lake Project used the right macroculture and the macroculture elements were very compatible with other social mechanisms that were in place. What the study means by the 'right macroculture', is the culture that is congruent with other social mechanisms and does not conflict with their interest. That is how macroculture becomes the most important of the social mechanisms. For example, in the current study, shared norms like avoiding the 'blame game', discouraging negative attitudes and voicing out issues became part of collective sanctions used in the Lake Project by the teams across locations and the other way around. The project team members were able to accept these practices as sanctions and used them continuously and repeatedly over time so that they became shared norms and part of macroculture of the Lake Project, because these sanctions were compatible with the Lake Project shared values and norms (macroculture).

Furthermore, the Lake Project continuously maintained the compatibility of their collective sanctions practices with their macroculture values. The Lake Project teams shared a culture of positivity, collaboration, and safe and trusting behaviours across geographic boundaries. They shared values of not imposing severe negative sanctions but promoting and rewarding positive attitudes. In addition, the Lake Project team members had a tendency to maintain the positivity within the project because one of their main and strong shared values was 'trust' and they were not willing to lose the project team members' trust by imposing severe negative sanctions that may not be compatible with their culture. Had they done such, it could have created complications in the coordination of the teams' interactions and maintaining safe and protected exchanges across borders, so they introduced the right sanction practices that were compatible with their current values and culture, such as avoiding the 'blame game' and discouraging negativity.

These sanction practices became shared norms amongst the project teams across locations and worked effectively. After a while, it was much easier for the teams across borders to accept them, use them over time, and they become part of the Lake Project culture as shared norms between the teams across geographic boundaries. It also reduced the amount of monitoring required and made coordination efforts easier across borders. The sanctions that are not normalised and do not become part of macroculture could be because the culture does not accept them or for other reasons. Therefore, macroculture and collective sanctions overlap.

Macroculture and Restricted Access

The second point noteworthy to pay attention is the interaction between macroculture and restricted access. The continued exchange amongst certain Lake Project team members across locations motivated team members to build trust (shared value) in their interactions and develop better mutual interests and strong ties, which are very critical for safeguarding exchanges in a distributed agile project such as the Lake Project. In addition, restricting access safeguards exchanges by increasing identification with the collective norms (macroculture) of the project because a smaller number of team members allows individuals to interact more frequently with each other, enhancing the sense of the team identity and shared goals (Sagers 2004). This allows stronger bonds to be formed amongst the teams, leading to better identification with the development team, increased trust and shared goals (macroculture) (Granovetter 1973).

I found that, through frequent interactions amongst fewer team members across locations, practices such as forums and developing common language become normalised over time. The project team members participated in forums and developed common languages frequently and these practices became norms amongst the team members across locations. In addition, the result of the study showed that some critical tasks became possible to be completed across geographic boundaries even though restricted access was applied to some project team members.

For example, in the Lake Project, due to some security reasons and the need to protect Ocean Group technology, team C in China had restricted access to part of the data despite the fact that access to the data was required to develop a critical part of the project. Because of high levels of trust shared among the project teams across locations, team C was able to follow the Australian teams to develop this major part of the project, which would have been impossible without the trust base among the team members in such distributed agile project. This example indicates that interactions exist between restricted access and macroculture.

This example indicates how macroculture and restricted access interact with each other. It also reminds us again of how critical is for firms to have the right macroculture practices in place that are harmonious with other social mechanisms in place. If a high level of trust was not part of the shared values amongst the Lake Project teams across geographic boundaries, it would have been extremely difficult to complete the project successfully and application of restricted access would have been more challenging.

Macroculture and Reputation

Macroculture and reputation can interact as well. It is also noted that macroculture is the most important and broadest of all social mechanisms and this is evident in its interaction with reputation. If there is a weak macroculture it is unlikely that there will be a strong reputation because the reputation of individuals is assessed in relation to those norms and values (macroculture). In this regard, Feller *et al.* (2008) argue that failure to continue to adhere to the current macroculture could result in damaged reputation/potential damage to reputation, the threat of exclusion from projects (collective sanctions), and the possibility of expulsion from the network (collective sanctions). These all serve to ensure that the project teams across locations behave as expected by the macroculture (thus facilitating coordinating exchanges), and that violators can be dealt with more effectively (thus facilitating safeguarding exchanges).

For example, in the early stages of the Lake Project, a technical leader in the Melbourne team was not adhering to agile practices although agility was part of the shared values of the project. This behaviour damaged her reputation and they had to remove her from the project. This example indicates the interactions amongst reputation, macroculture and collective sanctions and how they interact with each other. This also indicates that having the right macroculture in place gives firms greater ability to resolve exchange problems (coordinating and safeguarding). It also has an effect on the way other social mechanisms resolve exchange problems. If they are congruent with each other, especially with the prevailing macroculture, they have a better ability to resolve exchange problems.

Another example that indicates the interactions between macroculture and reputation is one of the social expectations (reputation indicator) in the Lake Project across geographic boundaries was the level of trust. Such social trust is identified by Lorenzen (2001/2002) as a shared expectation of honest behaviour in a community. When trusting behaviour became a shared value amongst the teams across locations, it encouraged positive reputation and behaviour amongst the project teams. The Lake Project teams were motivated to behave according to this social expectation to maintain their positive image.

Collective Sanctions and Reputation

If the team members do not adhere to the values of the project by not acting in a manner that is considered to be competent and professional, the consequences include damage to their reputation as well as the imposition of sanctions. This is consistent with the argument in the NG literature that the reputation of the team members would be damaged if they behaved unacceptably (Feller *et al.* 2008). In this regard, Hagen & Choe (1998) suggest that in social exchanges, conflicts, norms and values violations have to be resolved by collective sanctions. They further argue that in addition to resolving conflicts, this process also helps avoid conflicts, because when the project team members are aware of the effect on their reputation, they have a greater tendency to act more deliberately as expected. In this way, the teams' exchanges are also protected against opportunism.

In addition, the interaction of collective sanctions and reputation is pointed out by Lorenzen (2001/2002) where the author argues that reputation effects includes mechanisms of social sanctions (e.g. withdrawal of membership, reciprocity, loss of customers) towards those team members who violate the norms. When an individual is part of a community in which the likelihood of interacting with the other team members is relatively high, they will have incentives to comply with acceptable behaviours in order to maintain a good reputation.

Furthermore, Jones *et al.* (1997) claim that reputation has economic consequences (collective sanctions) for participants in NG. For example, in the IT industry, "Those with successful performances and track records move ahead in their careers, those with moderate reputations do not, those with poor reputations experience employment difficulties" (Faulkner & Anderson 1987:881).

Again, it needs to be pointed out that in order to be effective, all social mechanisms need to be congruent with each other. In the case of collective sanctions and reputation, if firms introduce sanctions that are not compatible with the project team culture, it will affect the reputation of the team members across locations. Incompatible sanctions reduce the project teams' motivation to collaborate and may lower their performance, thus impacting their reputation. As can be seen, the basis of harmony amongst social mechanisms is macroculture. When the right macroculture practices are in place, then collective sanctions have the compatibility with the culture and lessen the possibility of damaging team members' reputations. Thus, macroculture is one of the most important social mechanisms.

Collective Sanctions and Restricted Access

Like other social mechanisms discussed before, collective sanctions and restricted access can interact as well. They also need to be congruent in their application into the flexible and trusting environment of the Lake Project. For example, introducing incompatible and severe sanction practices, such as restricted Internet access or exclusion from social networks will very likely affect the team members' access.

Reputation and Restricted Access

There is also a potential interaction between reputation and restricted access. Because all four social mechanisms are interacting with each other, the same applies to reputation and restricted access. If you introduce sanctions that are not compatible with the project team culture it does effect the reputation and access for the team members across locations. Restricted access limits the number of the project team members that interact through relational contract. Incompatible sanctions reduce the project team members' motivation to collaborate and may lower their performance, thus effecting their reputation. On the other hand, introducing incompatible sanctions like restricted internet access or exclusion from social networks, in a flexible and trusting environment such as the Lake Project's, affects the team members' access as well.

It is also important to bear in mind that the relational contract values and norms were put in place based on the current macroculture. Therefore, once again macroculture plays a role in other social mechanisms' interactions. Relational contract membership encourages fewer team member to interact, but if the project team members do not act according to the contract rules and values, their reputation will be damaged.

In addition, strong embeddedness amongst the project teams enhanced flow of information about the team members' behaviour (reputation) across locations. Therefore, the team members are more likely to interact more frequently with the team members having positive reputations and attitudes and restricted their interaction with the team members with negative reputations. It is important to keep in mind that the reputations of the team members are assessed based on the current macroculture: What are their values? Are they trust worthy? What norms and goals do they share? Do they avoid blaming other team members? Do they encourage positivity? Here one more time, the role of macroculture has highlighted the influencing application and interaction of other social mechanisms.

From the preceding discussion, it would seem that there is sufficient evidence that these interactions may well exist and there would be a strong case for further investigation. In looking at these interactions and in conducting my study, it seems that macroculture is an important mechanisms.

The Importance of Macroculture

The previous section discussed how the four social mechanisms of NG interact. One key point that arose a number of times during the study was the extent to which macroculture was influencing the application and interaction of other social mechanisms. That is, not only do all social mechanisms need to be compatible with each other but it is most important to have social mechanisms that have harmony with the current macroculture. This point is alluded to in the literature as Jones *et al.* (1997) state that the congruent content of social mechanisms influences the coordination costs and safeguarding of complex, customised exchanges. They further argue that the more congruent the content of multiple social mechanisms for collaboration and sharing information, the greater the likelihood of NG emerging and thriving in rapidly changing markets for complex, customised tasks.

It is possible that this is one of the reasons that all social mechanisms in the NG model were able to effectively facilitate coordinating and safeguarding exchanges in the Lake Project. Throughout the project, all social mechanisms not only were congruent with each other but their practices were compatible with the existing macroculture. This compatibility amongst the mechanisms enhanced their interactions and therefore they were able to facilitate coordinating and safeguarding exchanges more effectively. For example, reputation and collective sanctions safeguard exchanges in NG by detecting and diffusing the team members' behaviours and high visibility of their activities. However, if they are not congruent with each other, the content of some social mechanisms may undermine others and create incoherence in the system. For example, macroculture content may inhibit and collective sanctions may penalise information sharing of honest information and undermine coordination, even when there is an appropriate social structure for spreading information about reputations.

As indicated above, in addition to social governance mechanisms interactions, I found macroculture to be the most important and the broadest of all the mechanisms. In order to have effective social governance mechanisms it has to be through the culture. This point is consistent with the NG literature that firms develop macroculture that results in a strategic and behavioral homogeneity amongst them because macroculture is a prerequisite for social control in inter-firm relations (Das & Tang 1998). It is also noted that in social exchanges, shared beliefs and values are particularly important for sustaining stable exchange relationships (Nord 1969). When exchange actors have common beliefs regarding their objectives and behaviour, the task of coordinating cooperative endeavours is less necessary and when it is necessary, it is easier and less costly (Das & Tang 1998).

One of the points highlighted while conducting the study is that macroculture seems to be very important and given the universality of macroculture (Hartmann & Gerteis 2005) and being a fundamental notion of all organisations, I expect that these points would have broad applicability. I would have expected that this applies to most organisations and in that they would find, in order to have effective social mechanisms that would be able to resolve exchange problems (coordinating and safeguarding) in the best possible way, they need to have the right macroculture practices in place, meaning the macroculture that is compatible with other NG social mechanisms. It is important that firms apply restricted access, reputation and collective sanctions practices that are congruent with the prevailing macroculture; otherwise they will not be able to facilitate coordinating and safeguarding exchanges effectively.

It has been noted that there is a strong macroculture existence in the case study and this enhanced the effectiveness of other social mechanisms. For example, in the Lake Project existence of strong shared values such as trust, encourages the project team members to have a positive reputation across locations in order not to lose other team members' trust. Thus, if you have a weak macroculture you cannot have a strong reputation, if the shared values like trust are not strong amongst the teams, then having a positive reputation would not be taken as important.

Macroculture is diffused and sustained through socialisation received in professions that shape decisional bases amongst distributed participants that creates strongly shared macroculture (Kaufman 1960, Van Maanen & Barley 1984). That could be the reason why the Lake Project established a strong macroculture (through right socialisation practices such as co-locating). Research studies in the international joint ventures literature also show high failure rates due to the difficulty of managing cultural differences amongst parties (Contractor & Lorange 1988). Therefore, having the right and strong macroculture appears critical.

While conducting the current study, I found a strong macroculture existed in the case study and this enhanced the effectiveness of other social mechanisms. Firms can have restricted access and collective sanctions but they are always being imposed while this is not possible for macroculture. Firms can make up rules for restricted access and collective sanctions but not for macroculture and values like trust, honesty and respect. If firms have an effective macroculture in practice, shared values, shared norms, shared goals and assumptions, then this in itself encourages a positive reputation and lowers the need for sanctions and a suitable amount of restricted access would apply. That makes macroculture the most important social governance mechanism. Therefore, in order to have effective governance it has to be through the culture.

Such a working environment reduces monitoring costs and opportunism besides enhancing coordination among the teams. For example, in the Lake Project the strong shared value of trust encourages the project team members to have positive reputation across locations in order not to lose the other team members' trust. Thus, if you have a weak macroculture you cannot have a strong reputation. If the shared values like trust are not strong among the teams, then having a positive reputation would not be taken as being important.

Feller *et al.* (2008) in their study of NG mechanisms in online communities, emphasised the significant role of macroculture and how important is to follow macroculture closely. They state that their study result indicates the participants believe that the culture within the network (rather than just restricting access) is much more important in safeguarding exchanges. They further argue that failure to continue to adhere to the prevailing macroculture could result in damaged reputation or exclusion from the project or the network collective sanctions. In addition, Bogg *et al.* (2010) note that central role of macroculture and the hypothesis that the lack of understanding of macroculture will likely result in ineffective exchanges and subsequently fail to resolve exchange problems because coordinating and safeguarding of exchanges will be more challenging and complicated.

In conclusion, macroculture is the most important and most critical with the widest context of social mechanisms that drives other mechanisms and their impact on coordinating and safeguarding exchanges. If the right macroculture is developed and the right practices are put in place, then resolving coordination and safeguarding exchanges through other mechanisms is much more achievable and effective.

6.8.2 Noteworthy business practices

One final section, while conducting the research and getting to know the organisation under study well, there were number of organisational practices that stood out. This section is a discussion of the points noted about the organisations' practices that I think worth noting as they might be useful to businesses.

While conducting an in-depth investigation and analysis into the Lake Project teams, I found several effective practices that are used in the Lake Project that the teams benefit from to enhance coordination and safeguarding across locations. One of the main practices is co-locating the distributed teams at the start of the project, and more importantly, repeating this practice through the project life cycle. This practice brought benefits of enhancing productivity, exchange of knowledge, building face to face relationships, encouraging trust, being familiarised with work culture and process, encouraging mutual interest and context awareness to the Lake Project teams across locations.

Indeed, the Lake Project faced similar challenges like every the other distributed project team that is not co-located and one of the main problems is lack of face to face communication. There is a direct inconvenience of not being face to face and not being very quick and responsive. It gets more complicated over time due to cultural differences; the team members do not feel part of the team because they are not always involved in everything. The Lake Project agile coach remembers *"I have had situations where if you don't put a lot of effort into the mitigating strategies, distributed team members tend to not feel like part of the team. If they don't get consulted, the team has to work hard and resolve the problems themselves and can feel even more isolated. Then if you don't do anything about it, it gets worse and worse."* However, the co-locating practice brought many major benefits to the teams in the distributed Lake Project that otherwise would not be possible. The following section explains the practice, the benefits and its impact on coordination and safeguarding of exchanges across geographic boundaries in more detail.

The Lake Project co-located teams and brought China and Melbourne teams to Sydney at the beginning of the project. Then they provided the teams with 20 different workshops, with every workshop lasting two to three days. During all these workshops, the Lake Project teams got familiar with what the context is and what the planning is to get started with iterations. The main goal of these workshops is to make sure that at the start of the project, the teams across locations are all on the same page and have the same contextual information before they go back to their original site. The Lake Project coach explains *"Having the team come here and be co-located at the beginning helped the coordination to improve."* Therefore, the coordination was enhanced among the teams and the project had a smooth starting phase with minimum distribution challenges.

The workshop and workshop planning helped the project teams to get started. But it can still be challenging because all they were doing was getting started – in the mean time they can be experiencing with problems. So the coaching is to get them started with the concept initiation phase and so they can have their plan in place before starting iterations. Then, the Lake Project agile coach is there to help with any problems the team members have. That in itself was very difficult because he was the only person and had responsibility of all streams for the project over all states. There was about 20 project streams for program A and one agile coach was responsible for getting them all started. Because of this, they had some issues in the Melbourne team, the team took longer to get started as there was only on agile coach and one program manager. Therefore, to overcome this problem and make coordination easier, the Lake Project provided teams with a big network support including agile coach, IM, senior business analysts and the team leader, which was very effective. Besides the co-locating workshops, the China team from the River had their own workshops running throughout the project. These workshops included technical courses, context courses and English courses, which enhanced the coordination and agility culture among the project teams across geographic boundaries.

Workshop participation is part of building knowledge so they can take the knowledge back to their location. However, it’s also part of building the relationships among the distributed teams while working together in these workshop sessions. It’s almost impossible to build these relationships among the team members from a distance if they never meet face to face.

That is why in addition to workshop sessions, during the co-location of teams from Sydney, Melbourne and China in the Sydney office, the team members participated in informal socialising sessions where they were playing games and chatting with other team members. During these socialisation practices, the team members from different locations that had never met before were able to break the ice and get to know the other team members personally. They were able to make friends and get familiar with the other team members’ behaviours, attitudes and personalities. This helped to build the initial trust and relational embeddedness at first stage of the project.

The Lake Project team had exchanges and co-locating practices all through the project. They had team members from China come to Sydney for two weeks to be part of the team and then go back to their location. Then a month later the team members from Sydney team went to China for two weeks. Similarly, the teams across locations were co-located in Melbourne for a period of time. *“We do that [co-locating] for the start of the project, but that is not enough and you have to keep doing that through the project. It’s a false economy to try save money by not doing that.”* (Agile coach, Lake Project) Although this practice involves costs for the travel, the benefits returned from this practice are of far more economic value to the Ocean Group.

For example, the Ocean Group had another project in Brisbane where they were also partnering with a team in China. All were very knowledgeable in agile practices but they did not co-locate teams and as the result the project had a lot of failures and challenges and did not go as well as the Lake Project. The Lake Project manager talks about this example: *“Our Brisbane project is evidence to show the huge advantages and lessons for the Ocean Group to learn about this practice and how well it paid off in later stages in the project. I think not having that in one of our current streams of program A is why we are struggling with the context.”* The co-locating practice helps the whole team make sure they are all on the same page when they start. The Lake Project was big but by co-locating the other teams in Sydney for a couple of weeks at the beginning of the project, they were able to build the strong ties among the teams and the availability of the contextual information helped to improve the coordination.

Australian teams experienced the enhanced coordination when they had the team members from Sydney and Melbourne in China for couple of weeks. It helped a lot in resolving some coordinating issues that had been very confusing before that. Having these members in China that understand Australian culture also helped coordinating exchanges. For example, when a team S developer went to China, she was coordinating things from there. She understands Australian culture and the way things are working in Australia. When she was in China, she was coordinating a lot of staff from there and she gave team C the context of why things are working this way, why these things are happening, why an Australian team member walked out of the workshop. So because she was on the ground there she could explain these things.

It worked the other way around as well because while there she could say to team C members that Sydney or Melbourne team members are not really clear about a particular subject that what they mean or explain Australian culture and logic. Because it's all about context and if the contest is not clear enough they are not able to find themselves in there or what reasons are in there. When they become aware of the context then it makes the coordination much easier for them.

It is very important to keep in mind that in order for co-locating practices to be effective you need to repeat this practice during the project. The reason that repeating the exchanges throughout the project is important is because the same benefits and understanding that is observed by team C when in Sydney would be obtained by team S in China when they co-locate with the other teams in China. The Lake Project IM explains: *"We wanted Australian team members to learn and understand how team C work together in China and how our ways of working here can impact that or how can we improve things here to make it easier for them or vice versa."*

The repeating of co-location helped the Lake Project teams a lot. *"We started to know the Chinese working environment and culture. Having six people out of 15 in Sydney and Melbourne going to China proved to be very helpful to understand their culture."* (Team leader, Lake Project) When Australian team members came back to Australia, they started sharing information on Chinese values and their way of working and it helped them to learn a lot. It helped them to understand why some of things happened the way they did and helped to coordinate teams in different locations.

Through repeating the co-locating practice, team S learnt and brought some of team C's practices to Sydney to benefit from them. For example, when team S were in China, they found team C had a huddle a few times throughout the day, like check points to check where the team members are at certain tasks. Team S brought this back to Australian teams. The other point that team S learned about team C while they were in China was that team C go to lunch exactly at midday. If they do not get the food on time their food court sells out – if they are not there by 12:30, everything is sold out. In Sydney, they had used to book the meetings at midday China time and team C used to get very upset and the Australian teams really didn't understand the reason because the food vendors would not sell out. After team S become aware of the problem they avoided booking any meetings at China lunch time, which helped the coordination a lot.

Another benefit of repeating the co-locating practice is that it helped the new team members who joined the project throughout the project life cycle. For example, there were two new team members from team C that came to Sydney during the project and spent at least four weeks in Sydney. During this time, new team members were able to build working and personal relationship with Australian teams face to face. That made the whole initiation phase for them and the other team members much easier, quicker and less challenging. Thus, they felt like a member of the team much faster and this enhanced their coordination with the other teams and reduced their incentives for behavioural uncertainty.

Finally, co-locating teams at the beginning of the project and taking them through workshops to educate them about the project context facilitates mutual adjustment by reducing the variances that project teams bring to their interactions. Mutual adjustment reduces variances in goals and expectations among the team members across locations. This leads to less monitoring thus facilitating safeguarding. It also makes the coordination among the teams much smoother across geographic boundaries.

In conclusion, having the co-locating practice at the start of the Lake Project and repeating it in different locations throughout the project life enhanced the trusting relationship among the team members and they were able to build a strong bond with each other. The strong relational embeddedness and trusting environment discouraged opportunism and behavioural uncertainty among the team members across geographic boundaries, thus enhancing safeguarding. The trusting environment among the team members also made the coordination much easier across all locations.

Other important practices that were found to be effective and have been discussed in previous sections in detail are buddy and shadowing. The Lake Project teams use these practices across locations to protect and safeguard their exchanges and to improve the coordination. During the buddy process, a new team member starts working on a task while they have a buddy. The buddy is usually a senior team member that sits next to new team member. They usually integrate quickly and get an understanding of what's happening so they can start bringing value and new ideas to the table. The buddy is there to support the new team member and check on him daily. A team S developer explains *"A lot of the time we tried to pair the new member up, so we find a buddy for them to introduce them to the project. You know it was really around let's give this person a piece to be responsible for and then guide them and support them through."* During shadowing practice, the new team member (shadower) follows one of the project team members dedicated to him (shadow) around to learn what he does day to day. It's a kind of learning by watching. Then after one week they switch, the shadow person will pull back and the shadower starts doing the work. When the shadow is happy with shadower's job, he backs off.

Buddy and the shadower team members are also additional sources to replace the team members that leave the project. Having these roles protects the other team members from possible opportunism because they are already part of the team, have built the trust and are familiar with the project team values, goals and norms. These practices also facilitate coordination because if these roles replace a missing role in the project, they are already part of the team and it is much easier to communicate and coordinate with the other teams across locations.

In addition to these practices, I found using a smart board like JIRA was very effective in providing transparency and visibility of the project teams' activities across locations. I found this very critical for agile distributed teams as it gives a clear idea to all the project team members the status of each task, avoids cross working, saves time and reduces monitoring efforts. The high visibility of the project teams' activities to the other project team members reduces the team members' motivation for behaving in an opportunistic way, thus enhancing safeguarding of the teams' exchanges and interactions across locations.

Finally, voicing out issues is another practice that enhanced the safeguarding of exchanges in the Lake Project across geographic boundaries. As discussed previously, the project team members are encouraged to speak out and communicate any issues or failures as soon as possible to avoid complications and opportunisms. The project team members repeatedly used this practice across locations so that it becomes a norm in the project. This practice made the coordination much less complicated because the issues were communicated before it was too late. It also promoted a 'safe-to-fail' environment for all the team members that protects their interactions across all locations. However, you have to keep in mind that the Lake Project was a high profile project and resourced from highly skill-full team members with good reputations. It is not certain that you will get the same result in every project, even with all of these practices. The Lake Project was the first distributed agile project for Ocean Group and put simply, there was so much effort put into it because failure was not an option.

6.9 Conclusion

In summary, I found that distributed agile development projects can be conceptualised as utilising NG and informal social mechanisms to coordinate activities and safeguard exchanges. Table 6.1 illustrates how the social mechanisms facilitate coordinating and safeguarding exchanges in distributed agile development projects. Indeed, the results of the current research are consistent with the NG literature regarding the fundamental application of social governance mechanisms. However, this case study result added new findings and prospects to the previous case studies. The key findings of this study are that applying Jones *et al.* (1997) NG framework in distributed agile development projects demonstrated that social governance mechanisms work differently in various contexts and their impact on coordinating and safeguarding exchanges is different also. In addition, I noted some points in regards to the relationships and interactions amongst social mechanisms. These relationships indicated the critical role of macroculture and its impact on other social mechanism applications and the way they can resolve exchange problems. Macroculture is now a broader construct.

Another noteworthy point is the importance of congruency across the social mechanisms. Developing the right macroculture gives firms more ability resolving exchange problems. It also has an effect on the way other social mechanisms coordinate and safeguard exchanges. If they are congruent with each other, in particular with prevailing macroculture, they have better ability to facilitate coordinating and safeguarding exchanges more effectively.

In conclusion, the case study allowed us to refine and delineate empirical indicators for the current study constructs and explain how social mechanisms facilitate coordinating and safeguarding exchanges (Table 6.1).

Table 6. 1 Social Mechanisms Impact on Coordination and Safeguarding Exchanges in Distributed Agile Development Projects

Social Mechanisms	Facilitate coordination	Facilitate Safeguarding
Restricted access	High clarity of roles	Centralised point of contacts
	High interaction frequency	Frequency of interactions (Strong embeddedness)
		Restrict access to certain members for certain type of exchange
		Membership in relational contract
Macroculture	Shared common language Shared common agile practices Shared norms to prevent fear of lock-in Shared common approaches to particular situation	Creating a 'safe-to-fail' environment through shared norms
	Unified team members' expectations across geographic boundaries	Creating a 'safe-to-fail' environment through shared values such as trust
	Shared focus on delivering software to the satisfaction of the business	
	Encourage new members to be compatible with the project culture	
Collective sanctions	Organising the project team members around agile values	Organising the project team members around agile values
	Encourage the team members to act in line with acceptable behaviours through high level of transparency and visibility of team members' activities	Encourage the team members to act in line with acceptable behaviours through high level of transparency and visibility of team members' activities
	Encourage the team members to behave in a manner that is regarded as competent and professional across borders through shared norms and reward acceptable behaviours	Encourage the team members to behave in a manner that is regarded as competent and professional across borders through shared norms and reward acceptable behaviours
Reputation	Resourcing project with members demonstrated positive reputation and competency	Resourcing project with members demonstrated positive reputation and competency
	High visibility of team members' activities encourage them to gain and maintain positive reputation	High visibility of team members' activities encourage them to gain and maintain positive reputation
	High level of trust between members	High level of trust between members
	Detecting and diffusing team members' behaviours	Detecting and diffusing team members' behaviours

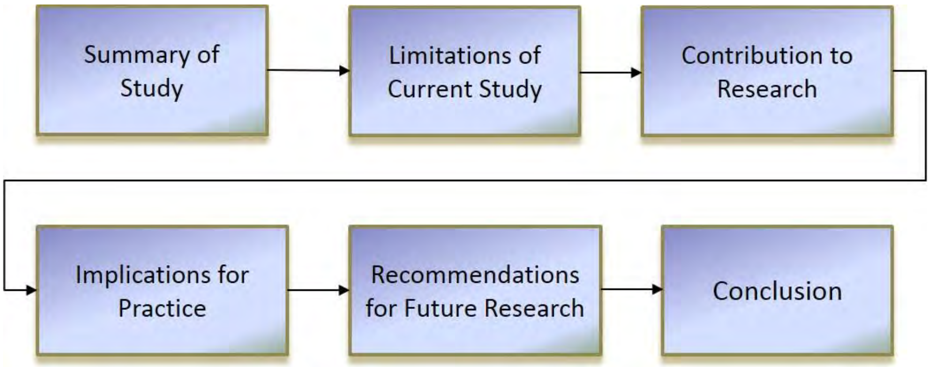
This study also provides a basis for future research on how distributed agile development projects are governed. Having a tested model of governance will allow the empirical exploration of different facets of success in distributed agile development context with a solid theoretical core. The next and final chapter talks about limitations of the current research study, provides recommendations for future research and how the current research advanced academic knowledge. Then it explains how the findings have impact on and reinforce the practice and businesses. Finally the chapter ends with a conclusion.

Chapter 7 Conclusions and Recommendations

7.1 Overview

This final chapter of the thesis commences with a summary of the study, followed by a discussion of the study's limitations, an outline of the study's contributions to theory, implications for practice, recommendations for future research and finally, a conclusion.

Figure 7. 1 The Structure of Chapter 7


Summary of Study
Limitations of Current Study
Contribution to Research
Implications for Practice
Recommendations for Future Research
Conclusion

7.2 Summary of Study

I sought to examine the potential benefits and challenges of adoptive social governance mechanisms in distributed agile development projects and suggested a list of effective practices for governance of those projects. The study also had the aim to determine what other, if any, mechanisms and practices are evident in distributed agile development project teams in the same context. Indeed, there were two major motivations that formed the basis of the current research. First was the pressing need to develop compatible and effective governance for distributed agile development projects (Ramasubbu & Balan 2008, Thomson & Vidgen 2013) and second was the need for effective practices for governance of development projects' challenges in a distributed context (Ravid *et al.* 2009, Thomson & Vidgen 2013).

In the recent literature, distributed agile development has received increased attention as much as the software community invites the scholars to extend their studies further with new solutions to distributed agile development challenges that enable development firms to generate and enhance value from their IT projects and activities (Ramesh *et al.* 2006). Research on distributed agile development governance is of importance since effective governance has been identified as the single most important predictor of the value that an organisation generates from its IT activities (Weill & Ross 2004). Thus, to benefit from their IT activities, development firms require an effective governance framework such as Jones *et al.* (1997) model with a set of control mechanisms to address distributed agile development project challenges. However, the current and previous studies did not use the Jones *et al.* model in distributed agile development context and seemed inadequate to explain this phenomenon with very little empirical studies.

Thus the research problem was set as, ***"How are distributed agile development projects governed effectively*?"** from which the following research questions emerged:

RQ1) How are social governance mechanisms applied in distributed agile development projects?
RQ2) How do social governance mechanisms coordinate exchanges amongst distributed agile development project teams?
RQ3) How do social governance mechanisms safeguard exchanges amongst distributed agile development project teams?

The research methodology selected for this study was an in-depth case study to validate and understand the research mode. It also allowed conducting an in-depth investigation and made better sense for practitioners by retaining the holistic and meaningful characteristics of real-life events. This case study is exploratory. In using an exploratory method, the study was able to clarify the complex aspects of applying social governance mechanisms including restricted access, macroculture, collective sanctions and reputation in distributed agile development projects and assess their impact on coordinating and safeguarding the project teams' exchanges. The primary instruments used to collect data include in-depth interviews, observational field notes and documentation such as project status report and the project core team activity report (Yin 2009).

The case study project investigated in the current research was the Lake Project, as part of a large program in the Ocean Group organisation. The Lake Project core team was distributed in three geographical locations, Sydney, Melbourne and China. The Lake Project was an online project with the aim of simplifying online policies and the legacy system within the Ocean Group. The Lake Project was the first distributed agile development project for the Ocean Group organisation and considered an important and high profile project for the organisation. Therefore, every effort was taken to ensure this distributed agile development agile project was successful. The project was well established in agile methodology as much as the agile culture influenced the interactive structure of the project that, in addition to HR (formal) interactions amongst the project teams (Appendix F, solid lines), there are also informal interactions between the teams (Appendix F, dashed lines).

In summary, I found that distributed agile development projects can be conceptualised by applying NG social mechanisms of the Jones *et al.* (1997) model to coordinate activities and safeguard exchanges. Indeed, the study confirmed all of the previous relationships in the Jones *et al.* (1997) model and previous literature in distributed agile development context, such as restricted access coordinates (Jones *et al.* 1997, Sagers 2004, Feller *et al.* 2008) and safeguards the project teams' interactions (Jones *et al.* 1997, Sagers 2004), macroculture coordinates and safeguards the project teams' interactions (Feller *et al.* 2008), collective sanctions facilitates coordination (Feller *et al.* 2008) and safeguarding of the project teams' interactions (Jones *et al.* 1997, Sagers 2004, Feller *et al.* 2008) and finally reputation safeguards the project teams' interactions (Jones *et al.* 1997, Sagers 2004, Feller *et al.* 2008).

In addition to the above relationships, I found the Jones *et al.* model is not complete and found a new relationship such as reputation, facilitates coordination of the project teams' activities across geographical boundaries, in the same way it safeguards the teams' exchanges.

Furthermore, this study added new findings to the previous case studies such as the context dependency of social governance mechanisms. I found that the four social mechanisms apply and work differently in various contexts compared to their application into other contexts in previous studies (Jones *et al.* 1997, Feller *et al.* 2008, Sagers 2004). Therefore, they have different impacts on coordinating and safeguarding of exchanges in different contexts, such as Open Source Software development projects.

Finally, I also found a new construct, 'ideological similarity' as part of application of restricted access in the Lake Project. 'Ideological similarity' is about a preference for more frequent interactions amongst the team members with similar interests locally and across geographical boundaries. This was more visible when, for example, Business Analysts (BA) or testers interacted through their own forum, BA's forum or testers' forum, because they all shared similar interests, skills and knowledge. Therefore, more frequent interactions amongst the team members with similar interests facilitate smooth interactions and enhanced coordination (Jones *et al.* 1997).

7.3 Limitations of the Study

While this study aimed to provide as reliable and valid result as possible, there are limitations to the current research. The results discussed above need to be considered in light of the following limitations. This section discusses the limitations involved in the case study approach, the participants and the techniques including interview instruments and techniques, transcripts and coding/interpretation.

7.3.1 Case Study Approach

This case study analysis, like any other case study, might be subject to a number of criticisms, the most common of which concern the inter-related issues of methodological rigour, researcher subjectivity, and external validity. With regard to the first point, the prototypical view here is that of Maoz (2002: 164-165), who suggests that, "The use of the case study absolves the author from any kind of methodological considerations. Case studies have become in many cases, a synonym for freeform research where anything goes." The absence of systematic procedures for case study research is something that Yin (2009: 14-15) sees as traditionally the greatest concern due to a relative absence of methodological guidelines. As the previous section suggests, this critique seems somewhat unfair; many contemporary case study practitioners have increasingly sought to clarify and develop their methodological techniques and epistemological grounding (Bennett & Elman 2010: 499-500).

A second issue, again also incorporating issues of construct validity, concerns that of the reliability and replicability of various forms of single case study analysis. This is usually tied to a broader critique of qualitative research methods as a whole. However, the latter obviously tend toward an explicitly acknowledged interpretive basis for meanings, reasons, and understandings. In this regard Berg and Lune (2012:340) suggest that, “Qualitative measures *appear* objective, but only as long as we do not ask questions about where and how the data were produced…pure objectivity is not a meaningful concept if the goal is to measure intangible [as] these concepts only exist because we can interpret them.” It is essential to develop a case study protocol in case study (Yin 1994 & 2003). The study presents the protocol as a major component in asserting the reliability of the case study research.

The question of researcher subjectivity may be intended only as a methodological critique of what are obviously less formalised and researcher-independent methods (Verschuren, 2003). Indeed, researchers can become a source of bias when they are the main instruments for data collection (Yin 2011, Goulding 2002). In this study, I was an outsider to the population being studied and had no direct contact or relationship with the organisation to be studied.

In addition, I employed certain measures to address the possibility of subjectivity. These were maintaining a high level of consciousness about these biases and remaining objective throughout the research, using peer reviews and the team member check (Lincoln & Guba 1985) to enhance the credibility of the study results (Creswell 2007, Goulding 2002), integrating collected feedback comments from selected participants into the report (Verschuren 2003), documenting the process of category development to foster openness and reducing any possible subjectivity (Creswell 2007), constructing the data team meetings and the minutes independently of this study to analyse them independently (Yin 2010) and finally, the connections and analysis made were solely from my viewpoint. In regard to the third point (external validity), to improve the limitations involved in this case study approach including the trustworthiness of the collected data (Russell *et al.* 2005) and establishing credibility (Guba & Lincoln 1994, Yin 2009, Trochim 2001), transferability (Eisenhardt’s 1989) and trustworthiness (Yin 2003), I used different strategies and techniques to improve these limitations.

With regard to trustworthiness, there are several basic key elements to this study design that have been integrated to enhance overall study trustworthiness as recommended in literature (Russell *et al.* 2005). First, I ensured enough detail was provided so that readers could assess the validity or credibility of the work. As a basic foundation to achieve this, I made sure that:

(a) The case study research questions were clearly written.

(b) The exploratory case study design was selected because due to the nature of the research questions for this study ("how" research questions), the case study design was the most appropriate (Yin 2014).

(c) Purposeful sampling strategies were applied (which were appropriate for the case study) (Patton 1990, Perry 1998) and different participants were selected to represent majority of the project team members. The participants were selected based on their involvements, skills and experience in the Lake Project.

(d) Data were collected and managed systematically; after each data collection session, the data were reviewed, clarified and confirmed in the follow up sessions. This process entailed reading the transcript of their interview, checking for correctness and completeness and providing additional information that they might had forgotten during the initial interview (Guba & Lincoln 1989).

(e) The data were analysed correctly using peer review strategy. The data were discussed and reviewed by two professional colleagues in the same field. Once the data collected and transcribed, it was discussed and reviewed by the senior academics who are familiar with qualitative methods. If I had any questions or doubt about clarity of the answers or coverage of the answers to research questions, they guided me through, commented and made some suggestions. Then the collected data were systematically coded (Section 4.5).

The case study research design used triangulation of data strategy to enhance data credibility. It supported the principle in the case study research that the phenomena can be viewed and explored from multiple perspectives. The collection and comparison of this data enhanced data quality based on the principles of idea convergence and the confirmation of findings (Knafl & Breitmayer 1989). Additional strategies that have been integrated into this study to establish credibility include the use of field notes and peer examination of the data (Guba & Lincoln 1989). Field notes were taken as information during the interview process. Notes also reminded me as the researcher, of the setting and sequence of occurrences, dialogue and events that transpired (Stainback & Stainback 1988).

At the analysis stage, the consistency of the findings or "dependability" of the data was promoted by, first independently coding a set of data and then working though that coding with experienced colleagues to come to consensus on the emerging codes and categories (Guba & Lincoln 1989). I also chose to implement a process of double coding where a set of data were coded, and then after a period of time I returned and coded the same data set and compared the results.

I followed a set of six criteria proposed by Miles and Huberman (1994) to evaluate the sampling and selection of the case for this study (Section 3.4) including:

- sampling strategy was relevant to the research conceptual framework and the research questions
- the selected case was able to generate rich information which were conceptually important for this study
- the selected case enhanced the generalisability of the findings, (d) the selected case produced believable descriptions/explanations;
- I considered the ethical nature of the relationship between researcher and informants;
- I considered the feasibility of the sampling plan

Following the above points, I selected the Lake Project because firstly, it was well distributed across four geographical locations including Sydney (NSW), Melbourne (VIC), Brisbane (Queensland) and Chengdu (China). Secondly, the Ocean Group could be considered a network because it consisted of independent subsidiaries with offices at various locations. In addition, the Lake Project met the four preconditions of the NG theory (Table 2.5) as explained in Section 4.3.

Another limitation with case study research is that any explanations from it may not necessarily be applicable to other situations. In other words, results from case study research are not empirically generalisable. However as Berg (2004, p. 259) states, “Few human behaviours are unique, idiosyncratic and spontaneous.” Generalisability was not the purpose of this study, however there are likely to be lessons from this study for organisations that are involved with IT projects and, in particular, those with distributed agile development teams. Future research may wish to look at the distributed agile teams in other contexts. Indeed, the purpose of the study was to flesh out how social governance mechanisms can be effectively applied in distributed agile development projects. However, the findings of this study may or may not applicable to other contexts (Barnes *et al.* 1994). It needs to take into consideration that the Lake Project was a special project, in a way that it was considered an important and high profile project within the organisation, therefore, the Ocean Group tried its best to source the project from highly skilled team members with positive reputations. However, this doesn't diminish the current study findings, it just means that the organisations need to understand how they may apply their own context.

To understand the phenomena of the study, the choice was made to investigate the phenomena in detail in a specific organisational context. I took an approach that limits how it applies to population. I chose one case for this study. In this regard, Walsham (2006) explains replicability is very difficult especially if the cases are unique to be studied or if the number of cases are very limited which will provide some limitation to generalise the result from the cases. However, Ragin (1992) explains that criticising single case studies for being lower ranked compared to multiple case studies is misguided, because even single case studies, "Are multiple in most research efforts because ideas and evidence may be linked in many different ways" (p. 225).

In addition, my approach was that explanations from case studies are generalisable to 'theoretical propositions' and not to 'populations or universes' and, as Yin (2003, p.10) argues, the case study does not, "Represent a sample and in doing a case study your goal will be to expand and generalise theories and not to enumerate frequencies." With a proper research design and precise measures, the current study aimed for the results and findings to be more generalisable. Mode of generalisation was theory-related analytic generalisation from one case.

In summary, I used different strategies to mitigate the limitations of the study. To enhance overall study trustworthiness I provided enough details for the readers so they can assess the validity or credibility of the research. The case study research design used triangulation of data strategy to enhance data credibility. I also used field notes and peer examination of the data (Guba & Lincoln 1989) to establish credibility. To improve the consistency of the findings, I independently coded a set of data and then reviewed the codes with other field professionals and colleagues to come to consensus on the emerging codes and categories (Guba & Lincoln 1989). I followed six criteria proposed by Miles & Huberman (1994) to evaluate the sampling and selection of the case for the current research study (Section 3.4).

7.3.2 Participants

The participants represented a wide range of people across the project and there was very little material differences to the answers that have been provided by interviewees and that includes at different levels and on different aspects of the project. I interviewed the range of people on the project from all of the different level and roles and given the high level of homogeneity amongst the responses, further interviews would have not made a difference.

I used a purposeful sampling technique. Participants for individual interviews were purposefully selected using the snowball technique (Miles & Huberman 1994) in which I contacted the project team members who had agreed to participate and who met the study criteria and asked participants who had agreed to be involved, to recommend fellow project team members to be approached. I selected participants from different teams with different roles to present and cover majority of the project team members.

However, because the snowball technique was used for identifying potential participants, it cannot be discounted that it is possible that there were different factions or cliques with particular perspectives within the project and one of these particular perspectives was predominant within the study's participants (Patton 2002). I do not consider this to be the case as over 70% of people involved in the Lake Project were involved in the study, including a number of senior managers. In addition, the frankness and range of responses suggests there was little bias or predominance of a one particular perspective amongst the study's participants.

7.3.3 Techniques

Interview instrument and techniques

The main data collection mechanism for this study was semi-structured interviews. Interviews are known to have a range of biases and are socially complex situations. Bias might occur due to a) poorly formulated questions (Kvale 1992), b) interviewees not giving truthful or complete answers (Collins *et al.* 2005), c) trying to present a particular invented picture (Weiss 1994) or d) participants distributed in different locations across geography.

In terms of controlling poorly formulated questions and to limit any potential bias, the instrument was tested via three pilot case study interviews as well as being reviewed by senior academic professionals during meetings. During these meetings, the interview questions were reviewed a couple of times. To assist the participant to be honest and open in their responses, anonymity and confidentiality was ensured to the participants, an interview protocol was used, and I prepared and conducted all interviews (Gable 1994). During the interview process, the interviewer made sure that if the participant appeared concerned about the questions, confirmation was asked.

In addition, to limit the possibility of getting invented pictures by the participants, the context of the study was explained to the participants clearly at the beginning of each interview session. Besides this, interviews were conducted with different team members that allowed confirmation of any particular examples given (Yin 2003). Furthermore, due to time differences and distance between the different locations that the participants were located, it was difficult to access overseas participants.

One of the issues in mixing local participants' face-to-face interviews with overseas participants' over the phone interviews, is that there can be different outcomes due to lack of face-to-face communication channels (Yin 2003). To control for this inherent bias, I sent the overseas participant a briefing/summary after the interview and asked if any clarifications were required. The interviewee then sent the confirmation of the brief to me. Given that considerable consistency where consistency was expected and diversity where diversity was expected, was evident across the interviews from all locations and levels within the project, it was felt that issues concerning the data collection process that can sometimes arise where different interview techniques are used, did not arise in this case.

To reduce the possibility of presenting a particular invented picture by the participants, I had intense exposure to the phenomenon under study so that rapport with participants was established, thus facilitating the collection of less biased data through the reduction of the potential for social desirability responses in interviews (Krefting 1991). As data were collected and analysed, I integrated a process of cross checking, where my interpretations of the data were shared with the participants, and the participants had the opportunity to discuss and clarify the interpretation, and contribute new or additional perspectives on the issue under study.

Transcripts

To control the fidelity of the transcripts, I listened back over the transcripts as the transcripts were read. When there were words that were difficult to understand, I didn't try to interpret them but reconfirmed the words with the participant (Poland 1995). I engaged the participants to respond to, and comment on, data and findings for their accuracy and credibility. If there were sections that were not covered or I needed more data to complete the analysing phase, the participants were contacted again and a follow up interview session was arranged.

Coding/Interpretation

In qualitative research, the aim is to understand and discover the phenomena using subjective data, which makes the research more interpretive and subjective (Strauss & Corbin 1998). Therefore, this enabled me as the researcher to have an in-depth and detailed understanding of governance mechanisms that are practiced in distributed agile development projects, using the subjective information. I was mindful when doing the coding, of the potential for coding biases. The coding I used for this study was interpretive coding, which can be accused of being subjective (Walsham 2006). Indeed, researcher-bias is possible due to data analysis procedures and subsequent interpretation of the transcribed interviews (Kvale 2007, Creswell 2007, Weiss 1994). The interpretivist paradigm can create the subjectivity bias for this study, meaning the data analysis might have been biased by the researcher's personal beliefs, subjectivity, and sense-making.

To mitigate this limitation, after I did the coding, it was discussed with a second researcher who did not participate in the coding. I also left the coding and then came back to it later to make sure the coding was done properly. In order to maintain coding consistency, I reviewed previous transcripts and the coding was reviewed against current transcripts. There were certain parts where I was not sure what they meant, so I used the follow up technique and mentioned them in the follow up interview sessions with the interviewee, checked the content and made sure of the exact meaning. The technique used in the coding is similar to the technique suggested by Miles & Huberman (1994).

Another strategy used to reduce the subjectivity was that after each interview I wrote interview notes and discussed and compared the notes with the interviewee the next day. Also, to maintain a chain of evidence, a case study database was used (Lincoln & Guba 1985). Hence, these techniques and controls should add to the credibility of the study and indicate the results and findings are acceptably relevant, and appropriate to answer the research questions asked.

One of the most prominent concerns with case study research is its alleged lack of rigour (Yin 2003). It is argued that often case study researchers do not follow systematic procedures and that biased views influence the findings and conclusions of the study (Yin 2003). The omission of a discussion on validity issues in most qualitative research often reinforces criticising the method (Marschan-Piekkari & Welch 2004). However, Campbell (1975), Geertz (1995) and other scholars have shown that this critique is incorrect, because the case study has its own rigour, different to, but no less strict than the rigour of quantitative methods. The advantage of the case study is that it can reach real life situations and test views directly in relation to phenomena as they unfold in practice.

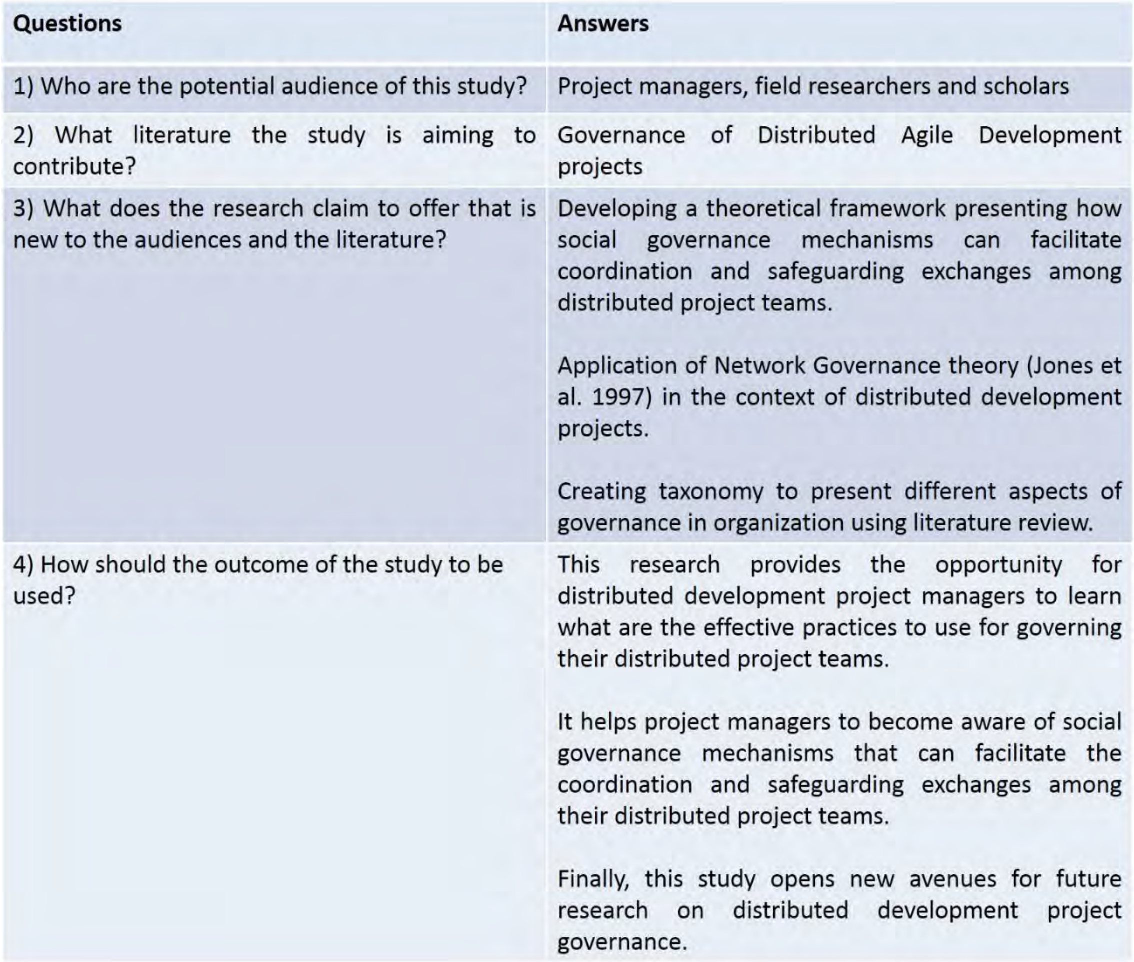
According to Campbell (1975), Ragin (1992), Geertz (1995), Wieviorka (1992), Flyvbjerg (1998, 2001) and others, researchers who have conducted intensive, in-depth case studies typically report that their initial views, assumptions, concepts, and hypotheses were wrong and that the case material has forced them to revise their hypotheses on essential points. To overcome this limitation, after collecting and analysing the data, the study material was reviewed a couple of times. This process resulted in having a revised governance framework at the end of the study.

In addition, the terms and definitions involved in the study topic might be difficult for some people to understand such as culture, values, assumptions and sanctions. I was very mindful of this when conducting the interviews to ensure that the interviewees understood what I was asking, and then I looked to understand their definitions and perspective. The interviews started with a section where the definitions that were used in the interviews were discussed and as part of the interview process I looked to ensure that the interviewees understood any terms used by the participants.

7.4 Contributions to Research

There are three main contributions of the current research: (1) advancing knowledge in terms of confirming, and adding to, the current NG framework, (2) advancing knowledge in terms of current practices, including the documentation of the unique circumstances of the project being studied and (3) the research method. The following sections address each of these contributions in turn. Following Walshman's (2006) guideline, the study answered four major questions to explain the potential contribution of the study (Table 7.1).

Table 7. 1 Contribution of the Study



Questions	Answers
1) Who are the potential audience of this study?	Project managers, field researchers and scholars
2) What literature the study is aiming to contribute?	Governance of Distributed Agile Development projects
3) What does the research claim to offer that is new to the audiences and the literature?	Developing a theoretical framework presenting how social governance mechanisms can facilitate coordination and safeguarding exchanges among distributed project teams. Application of Network Governance theory (Jones et al. 1997) in the context of distributed development projects. Creating taxonomy to present different aspects of governance in organization using literature review.
4) How should the outcome of the study to be used?	This research provides the opportunity for distributed development project managers to learn what are the effective practices to use for governing their distributed project teams. It helps project managers to become aware of social governance mechanisms that can facilitate the coordination and safeguarding exchanges among their distributed project teams. Finally, this study opens new avenues for future research on distributed development project governance.

7.4.1 Advancing Knowledge

The major contributions of this study are three-fold. Firstly, the study demonstrates that the Jones *et al*. (1997) model of NG social mechanisms is applicable to the distributed agile development project context. Secondly, it uncovers a new relationships within the model (Orange arrow in Figure 6.2). Finally, it finds a new construct that contributes to the application of restricted access, ‘ideological similarity’. In making these three major contributions the study confirms and validates the existing relationships in the Jones *et al.* (1997) model (Blue arrows in Figure 6.2) and the relationship confirmed by other scholars in literature (Green arrows in Figure 6.2) and in fact it does confirm them, as was expected. In addition to these major findings, the study also makes a number of other findings that should be of interest to researchers in regard to how social governance mechanisms are applied and how they resolve exchange problems in a distributed agile development context.

Jones et al. Model Applicable to the Distributed Agile Development Context

This study set out to understand that social governance mechanisms that operated within a distributed agile development project environment, make use of the well-established model developed by Jones *et al*. I found unequivocal empirical support for the model. The study results confirmed the following Jones *et al.* (1997) model of NG social mechanisms’ relationships and indicates that the Jones *et al*. model is applicable to the distributed agile development projects:

- Restricted access coordinates the project teams’ activities (Jones *et al.* 1997, Sagers 2004, Feller *et al.* 2008)
- Restricted access safeguards the project teams’ interactions (Jones *et al.* 1997, Sagers 2004)
- Macroculture coordinates the teams’ activities (Jones *et al*. 1997, Feller *et al*. 2008)
- Macroculture safeguards the project teams’ interactions (Feller *et al.* 2008)
- Collective sanctions facilitate coordination of the project team members’ activities (Feller *et al.* 2008)
- Collective sanctions safeguard the project teams’ interactions (Jones *et al*. 1997, Sagers 2004, Feller *et al*. 2008)
- Reputation safeguards the project teams’ interactions (Jones *et al*. 1997, Sagers 2004, Feller *et al.* 2008)

I undertook an in-depth analysis of the governance structures of the Lake Project using the Jones *et al.* (1997) model and found the model is sound and works as a basic structure in distributed agile development context, but the way it applies and works in such context is different compared to other contexts. This is in agreement with the literature. For example, the Jones *et* *al*. model was examined in other contexts such as OSS networks (Feller *et al*. 2008, Sagers 2004) and it was found that the ways in which OSS networks overcome exchange problems contrasts with other contexts.

The current research study contributes to the NG theory because it demonstrates how interactions can be coordinated and safeguarded in networks in which participation is not entirely voluntary. While prior research studied voluntary networks, such as OSS (Feller *et al*. 2008, Sagers 2004), in which interactions and particularly contributions depend on the goodwill of the team members; in a corporate distributed agile development project, setting entry to the network is gained through employment and from there on interactions, contributions, and delivery are expected by management and customers.

New Relationships within the Jones et al. Model

In addition to the above relationships, one new relationships emerged (Orange arrow Figure 6.3) as follow:

- Reputation facilitates coordination of the project teams' activities across geographical boundaries. I found that this relationship emerged in the same manner as reputation facilitates safeguarding exchanges across locations, through practices that encouraged the project team members to maintain their positive reputation, thus there were more tendencies to coordinate. This relationship has not been suggested in the literature.

The current research study also provides empirical exploration of different facets of governance in distributed agile development projects, particularly in agile settings. As part of the additional noteworthy points (Section 6.8), I noted that four social governance mechanisms interact but the interactions could be quite complicated because the interactions between the social mechanisms are not mutually exclusive. Whilst some authors have touched on this point (Schroeder 1995, Feller *et al.* 2008, Jones *et al.* 1997, Lorenzen 2001/2002) briefly, I examined each interaction in turn, and demonstrated that there are interactions amongst the four social governance mechanisms and found they are consistent with the literature. I have gone through it thoroughly, systematically looked at the relationships, talked about it, put them together and created a much better understanding. Indeed, I followed what Jones *et al.* (1997) suggest is required as the research stream for examining the interaction of social mechanisms that could provide important insights. Jones *et al.* (1997) further suggest that we must have a better understanding of how social mechanisms reinforce, substitute or undermine one another and how their combination influences performance. Therefore, it is worthwhile doing what Jones *et al.* (1997) suggested.

Another noteworthy point discusses in section 6.8 is highlighting the importance of macroculture and how macroculture influences the application and interaction of other social mechanisms. Macroculture and its content and development is one of the main areas that Jones *et al.* (1997) identified for further research in NG because the content of macroculture in NG is poorly understood. In addition, the studies (Jones *et al.* 1997, Feller *et al.* 2008) suggest that there is a need for further research to understand whether some social mechanisms are more important. While I noted macroculture was important but I was unable to determine the scale of importance and how it came about. Further research might wish to pursue this.

New Construct 'ideological similarity'

'Ideological similarity' is a new construct, which emerged from the findings that were used in the case study for application of restricted access. The result of this study indicates that the tendency of the project team members to have more frequent interactions with the other team members that were ideologically aligned across locations encouraged restricted access.

Agile Development Context

The study also contributes to the body of knowledge in agile and distributed agile software development projects by presenting a comprehensive case study analysis of how social governance mechanisms are applied and resolve exchange problems in a distributed agile development context, hence leveraging the knowledge of field researchers and scholars. For example, the study explains in details how macroculture is applied considering agile values and norms in a distributed agile development context. The study also provides examples of how collective sanctions are applied in an agile distributed context while having agile values and norms as part of macroculture in the project.

Indeed, the study is a unique contribution to the literature because it is probably one of the few case studies, as far as I am aware of, that applies NG framework into an informal governance structure within a formal governance structure. Poppo and Zenger (2002) state the importance of having the informal (social) and formal governance and suggest that formal governance and informal governance function as complements. In the current research study, the Ocean Group is a corporate setting, shaped by a commercial aim and with formal managerial procedures in place. At the same time, social governance mechanisms applied to the Lake Project with informal structure (distributed agile software development project) where the culture of the agile development team is one of passion and engagement. The study also advanced knowledge in terms of current practices, which helps future researchers looking at the distributed agile development projects, to understand what they need to research (Dingsøyr *et al.* 2012). Abrahamsson *et al.* (2009) state that more research is required to gain a deeper understanding of how agile methods are deployed in practice, and is an overall necessity to improve the level of rigour in agile system development research.

The study clearly articulates an example of a large and complex distributed agile development project, which will benefit and inform the researchers that are interested in these projects. The study informs future researchers about the current practices in the real world. For example, I found that different agile methods that are categorised in academic papers are not practiced the same way in practice. The organisations mix and match different agile methods' practices, such as XP and Scrum practices, the way they match and are compatible with their project and organisation structure and settings. There is no pure XP or Scrum in the practical world of IT organisations and they are usually used as combined practices and researchers need to be mindful of this.

7.4.2 Unique Circumstances

The current research, studied a project case with unique circumstances. The Lake Project was the first agile distributed agile development project in the Ocean Group and to make it successful, they put in a lot of effort to source the project with the advanced team members with a good reputation. The project was an important and high profile project for the Ocean Group organisation. They also made some updates in project structure, like adding additional points of contact, to act as coordinators for such a large agile distributed project. In this regard, Mattsson *et al.* (2010) explain the importance of distributed agile development projects and their challenges. Indeed, this study has articulated how large distributed agile projects can be governed effectively applying social governance mechanisms properly. The study results also indicate that under unique circumstances such as the Lake Project, where the project is resourced with skilful and highly reputable individuals and effective strategies were put in place (i.e. co-locating practice, additional point of contacts, the project and HR reporting systems, 'THANK YOU' system), distributed agile development projects can benefit from the social governance model of Jones *et al.* (1997) to resolve the exchange problems effectively.

7.4.3 Method

Researchers have had trouble understanding agile techniques in distributed agile development teams (Dingsøyr *et al.* 2012). Despite the benefits obtained through distributed agile development projects, there are many difficulties faced by various organisations caused mainly by geographical boundaries and distance, time difference, communication challenges and cultural differences (Shrivastava & Date 2010).

One of the key contributions of this study is that it demonstrated an approach that could be taken to investigate a large case study project with the teams distributed across geographical boundaries. Different phases of conducting the research described in the study (defining and bounding an emergent and observable phenomenon, refining the preliminary model through qualitative case study analysis) proved to be a highly effective and useful process in addressing a social phenomenon, which requires both broad and deep analysis. The approach taken in this study was a comprehensive approach starting with getting access to the case study organisation, spending time talking to different participants in the organisation and understanding the project structure (Miles & Huberman 1994). Then I collected the data from the participants and went back and talked to them in follow up sessions. Indeed, part of the study method was to gradually build up a profile of the project, identify all the project team members in the project and find out who are the team members that I need to talk to.

The current study therefore calls for comprehensive case study research on social governance mechanisms in overcoming exchange problems in both distributed agile development and in other contexts. In addition to calling for replication studies, this study has implications for how researchers should conceptualise their studies of social governance mechanisms.

7.5 Implications for Practice

The study contributes to practice by finding effective governance practices in distributed agile development projects, relationships in social governance mechanisms and a revised governance model for practitioners that confirms the implications for businesses that practitioners need to be aware of them. One of the implications for practice is the lessons learned and how the study findings will change the way this area should be practiced. In light of the current study findings, a number of implications were found that practitioners (including development firms and the project managers) need to follow. The key ones are discussed in the following paragraphs.

7.5.1 The Structure of the Project

A key implication for practice is the importance of the structure, because the structure of the project drives the social governance mechanisms. For example, co-locating distributed agile teams throughout the project lifecycle is very critical for an effective governance and enhanced distributed agile development team. The co-locating practice and its benefits is explained in details in the next sections. Coordinator roles in the project structure are strongly recommended especially for large distributed agile projects to coordinate tasks across locations, save time and effort and lessen interruptions and confusion in the project. The current study findings suggest that IT organisations need to have both, formal and social governance mechanisms in place in order to benefit from an effective governance and cannot ignore one or the other.

Another critical point here is that IT organisations need put in effort to increase visibility of the teams' activities in distributed agile teams. This would encourage them to gain and maintain positive reputation. I also recommend that the organisations need to make sure the team members are very well aware of the importance of the project because this would be a very good motivation for them to gain or maintain positive reputation, which can improve coordination of their exchanges across locations. Indeed, for a professional, reputation is everything and the project team members distributed agile development environment will do whatever it takes to maintain that reputation if they want to continue working in that and other corporate networks. Reputation also emerges at the team level and plays an important role in group identity and cohesion – the team members that jeopardise the team's reputation will be subject to collective sanctions.

7.5.2 Consideration of Social Governance Mechanisms

The current research shows that management should pay attention to the impact of the four social mechanisms and utilise them effectively to resolve exchange problems in distributed agile development environments. One of the key things is that practitioners often think they need to focus on each social governance mechanism individually. However, the case study indicates that you need to focus on their interactions and how they are applied. It is critical that you manage all four social mechanisms and have a good understanding of how closely they interact with each other (Section 6.8). You need to not only know they all interact with each other but also how they influence the application of one another. You also need to know that they have to be consistent and concurrent with one another to function effectively and the businesses are advised to be focussed on that because congruent mechanisms reinforce one another to promote cooperation (Jones *et al.* 1997).

For example, if you introduce a sanction that is not compatible with the project team culture it does effect the reputation and access for the team members across locations. In other words, if you constrain one you can lessen the other's meaning. If you have strong trusting value as culture you would not need strong restricted access and severe collective sanctions and thus promote gain and maintain positive reputation amongst the team members because the project team members are not willing to lose the trust of the other team members. The way the culture is set can have implications on how the other social mechanisms are applied and the way restricted access, collective sanctions and reputation are set, also effect macroculture.

For example trust was a strong shared value amongst the Lake Project teams and that is why they tend not to impose extreme and negative sanctions for unacceptable behaviours. This is because it did not have the harmony of the project team culture and would have created complications. Instead, in order to maintain the trusting culture amongst the teams, they promoted positivity and encouraged and rewarded positive attitudes to become a role model for others to map the same behaviours.

Findings of the current research study introduce ideological similarity as a new construct for restricted access. Businesses need to be aware of the notion that people tend to talk with each other in NG. From a business perspective this would appear to be both detrimental and beneficial on the basis that it could create some sort of exclusivity and close down communication unnecessarily or could be beneficial because it increases frequency of interactions amongst certain team members with similar ideology across locations and enhance the communication between them. Ideological similarity is something that businesses need to be mindful for.

Previous studies did not find any relationship between reputation and coordination; however, the study findings suggest practitioners need to know that reputation can facilitate coordination of exchanges in distributed agile development projects. They can enhance coordination of the project teams' interaction through high levels of trust, sourcing the project with members that demonstrated positive reputations, detecting and defusing the teams' behaviours, high visibility of the teams' activities and make the teams aware of the project importance.

Importance of Macroculture

As part of noteworthy points of this study, the current research suggests that businesses place their focus on macroculture as it was noted to be the most important social governance mechanism, with the broadest context, and is more complicated than others (Section 6.8). Hence, it is important that managers properly understand macroculture. Firms need to be aware that in order to implement an effective social governance mechanism, they need to do it through the macroculture. You need to be aware that not only do all social mechanisms need to be compatible with each other but it is most important to have social mechanisms that have harmony with the current macroculture. I would have expected this applies to most organisations. Practitioners need to understand, in order to have effective social governance mechanisms that would be able to resolve exchange problems effectively in the best possible way, it is critical to have the right macroculture practices in place that are concurrent with other social governance mechanisms and the project goals. This is because lack of understanding of the central role of macroculture will likely result in ineffective exchanges and subsequently fails to resolve exchange problems.

Other social mechanisms such as restricted access, reputation and collective sanctions are usually imposed while this is not applicable for macroculture. Firms may be able to make rules to impose restricted access and collective sanctions but they are not able to enforce trusting values and respect. However, when restricted access, reputation and collective sanctions become normalised, they become part of the culture and self-correcting. This encourages positive reputation, less sanctions and a proper amount of restricted access. Therefore, there is less effort required to apply and constantly monitor social mechanisms because they function more effectively and easier. Although practitioners need to keep in mind that the Lake Project was a high profile project and it had a lot of writing on it and worked very well. This suggests that practitioners may need a lot of effort to be put in, to have the right macroculture in place.

Noteworthy Business Practices

Considering the above implications for practice, some practices were observed in the project that made the project successful. The participants indicated these practices as being important and very effective in social governance of distributed agile development projects. For firms seeking to meet challenges of governing distributed agile development projects, the study sees practices such as co-locating distributed agile teams, high level of transparency and visibility, additional points of contacts, buddy and shadower roles and voice out issues, as actionable implications of the current study. These practices enable IT organisations to implement the NG model that effectively govern distributed agile development projects. The practices are explained in details as follows:

Co-locating: in practice, firms are often unable to have the entire development team co-located and distributed agile development teams face challenges around communication, lack of control, and lack of trust. To overcome these challenges, companies need to facilitate knowledge sharing, improve communication, build trust, enhance context knowledge, verify outcomes and continually adjust the process (Ramesh *et al*. 2006). Based on the current research findings, it can be recommended that companies, through frequent co-locating of distributed agile teams at the beginning of the project and by repeating this practice throughout the project phases, will be able to implement the above points. In doing so, firms are more likely to retain or gain agility in a distributed agile development environment because all these are areas that agile software development has typically been strong in.

This practice brings the benefits of enhancing the productivity, exchange of knowledge, building face to face relationships, encouraging trust, getting familiarised with working culture and processes on different sites, encouraging mutual interest, building strong ties amongst the teams and finally context awareness and availability of the context information to the project teams across locations. Therefore, IT organisations will have improved coordination and experience a smooth start to the project phase with minimum distributed challenges involved.

Based on this study's findings the above benefits are achievable through workshop participation during co-locating practice. I recommend practitioners to provide the team members with different workshops because workshop participation is part of building the knowledge, the relationships and consecutive trust amongst distributed agile team members while working together in workshop sessions. In addition to workshop sessions, I encourage practitioners to provide the team members with informal socialising sessions during the co-locating practice, where they get familiar with the other team members' behaviours, attitudes and personality so they can benefit from building the initial trust and relational embeddedness at first stage of the project.

A key point about co-locating distributed agile teams is that it is very important that IT organisations keep in mind, in order for co-locating practice in different sites to be effective, the organisations need to repeat this practice during the project life cycle. The reason is that through repeating the practice, the same benefits, understanding and lessons learned, observed by one team (for example working culture), would be observed by the other team too. Also because each time, only a limited number of the project team members are able to participate in co-locating practice, by repeating the practice, the organisations are able to provide this opportunity for the other team members as well. This is especially important for new team members joining the project, because during this time they are able to build working and personal relationships with the other teams face to face and feel like they are part of the team easier and quicker. Therefore, initiation phase for them and the other team members would be less challenging. In this way the IT organisations are able to enhance existing and new team members' coordination with the other teams and reduce their incentives for behavioural uncertainty.

Although this practice involves economic costs for travelling, the benefits IT organisations get from this practice is far more economic that they are better off having it rather than ignore it. In this regard, the case study project agile coach explains that, "…we do that [co-locating] for the start of the project, but that is not enough and you have to keep doing that throughout the project. It's a false economy to try save money and not do that."

High level of transparency and visibility: practitioners need to know that the use of social mechanisms in DD projects, in order to overcome exchange problems, is only possible if the activities of the project teams are persistently visible to the rest of the network. I suggest that the transparency and visibility of the project teams' activities across borders is a critical part in social governance of distributed agile development projects. The more effort you put on visibility, the less monitoring costs are involved and more trusting relationships are built. By providing transparency and high level of visibility of the project teams' activities across locations, you can give a clear idea to all the project teams across locations about the status of each task, avoid cross working and confusion, save time and reduce monitoring efforts and lessen team members' motivation for opportunism. The tool used in this case study to improve visibility was JIRA which worked very well.

Additional points of contacts: based on the current research findings, to improve coordinating exchanges across locations, I recommend practitioners to introduce additional points of contacts to the project structure for large distributed agile development projects. These additional contacts have coordinator roles amongst the teams, which reduces work interruptions, lessens communication and workloads for each team member, and makes the communication and coordination less confusing and more focussed.

Buddy and shadower roles: Other important practices that are found to be effective are buddy roles and shadowing. Practitioners are advised to use these practices across locations to protect and safeguard exchanges and to improve coordination amongst the project teams. Buddy and shadower team members are additional sources to replace the team member that becomes unavailable during the project. By adding these roles to the project, you can protect the other teams from possible opportunisms because these roles are already part of the team, have built the trust and are familiar with the project team values, goals and norms. By introducing these roles to the project structure, you can also enhance coordination because they are already part of the project team and communication and coordination with the other teams across locations is much easier.

Voice out issues: I advise practitioners to encourage the project team member across locations to speak out and communicate any issues or failures as soon as possible. Distributed agile development projects can greatly benefit from this practice because it enhances safeguarding of exchanges by promoting a 'safe-to-fail' environment for all the team members thus reducing opportunisms. You can also improve coordination of exchanges across geographical boundaries because voice out issues early avoids coordination complications. Finally, the above recommended practices worked very well in the case study and the practitioners need to keep in mind that the Lake Project case study was a high profile project and resourced from highly reputable and skilful team members. Therefore, in order to benefit from these practices in the most effective way, I advise practitioners to put a lot of effort in resourcing the project team members and structuring their distributed agile development project.

7.5.3 Selection of Project Members

It is important to source the project with competency and team members with a positive reputation and make sure that the team members are well informed about the other team members' reputation and trustworthiness. This makes it much easier for the project team members to trust them more to fulfil their responsibilities and their ability to perform the tasks, thus enhance coordination amongst them.

7.6 Recommendations for Future Research

The current study identifies opportunities ripe for future research efforts in NG. Several scholars have noted that research on NG is increasingly important and became more prevalent because the exchange conditions — uncertainty, human asset specificity, and complex tasks — are increasing (Daft & Lewin 1993, Volberda 1996, Jones *et al.* 1997, Torfing 2005, Lewis 2011). Indeed, research into governance networks is offering important opportunities for theoretical and methodological development, and for the generation of new knowledge with both academic and policy relevance (Torfing 2005). In addition, work has increasingly shifted to knowledge-based modes where human asset specificity and the transfer of tacit knowledge across boundaries are important.

Research on NG is not only of theoretical but also of practical importance (Jones *et al*. 1997). The practical implications of the current study highlight the risks for those practitioners that might seek to use NG mechanisms without considering their interactions and requiring congruency of social governance mechanisms. Without considering these points, both coordination and safeguarding are likely to suffer. Nonetheless, there is still much to learn about social governance mechanisms' interactions and required compatibility amongst them. There is a need to have a better understanding of how social mechanisms reinforce, substitute or undermine one another and how their combination influences performance.

This area exceeded my expectations regarding its complication. Therefore, I suggest that two areas for further research are required. One research stream on examining the interactions amongst social governance mechanisms, which is expected to provide important insights because social mechanisms are critical to networks functioning effectively. The other one on their concurrency in distributed agile development projects and other networks. The study revised model of Jones *et al*. (1997) provides an enhanced understanding of how the four social governance mechanisms resolve exchange problems. It also provides guides needed for further empirical research on NG of distributed agile development projects.

In addition, as part of noteworthy points highlighted by the research of this study, macroculture considered as the most critical and broadest social governance mechanism. It also greatly influenced the application of other social mechanisms and their impact on coordinating and safeguarding the teams' interactions. Macroculture is a broader and more complicated context than what has been discussed in literature (Ziggers *et al*. 2010, Steffens *et al.* 2004, Sagers 2004, Jones *et al.* 1997). I recommend more research is required on macroculture, its content and development in NG. Further research is needed to investigate how practitioners can build an effective macroculture in order to have effective social governance. Future research might also investigate ideological similarities, its functionality and impact in NG in more details. This is a new and interesting construct that merits research on its own. Indeed, this study provided a starting point, but more work is needed in this area. There could be other factors that are required to be introduced to provide more insights into the relationship between social governance mechanisms and coordination and safeguarding of exchanges.

The noteworthy points of this study suggest effective practices that are important and work very well in NG of distributed agile development projects. However, introducing other practices may provide more insight. Therefore, I recommend that another research area would be to find out other effective practices that work effectively in NG. Another extension of this research could be the development of testable and measurable hypothesis as suggested by Eisenhardt (1989) to enable further theory building from case studies. Such work would be quantitative and allow for enhanced explanatory power of social governance mechanisms in distributed agile software development projects. I also call for multi-method research on social mechanisms in overcoming exchange problems in both distributed agile development projects other business networks as well.

Finally, the study results indicated that NG model is shaped differently across different industries as previous studies suggests (Jones *et al*. 1997, Sagers 2004, Feller *et al.* 2008). Therefore, I suggest effectiveness, similarities and differences in functionality of NG mechanisms by comparing NG models in divergent industries such as the IT industry (e.g. online communities, distributed and co-located agile software development projects), film industry and the industries that social governance mechanisms are applicable.

7.7 Conclusion

This study has demonstrated that NG theory is appropriate for the study of distributed agile development projects, particularly when agility is practiced. More specifically, this study sought to understand how the four social mechanisms contribute to the coordination and safeguarding of exchanges amongst distributed agile development project participants. The Ocean Group case suggests that all four mechanisms are relevant in a distributed agile context and that, separately and jointly, they contribute to coordination and safeguarding. I suggest that social governance mechanisms are not independent and interact with one another and this has impact on their application and functionality in a distributed agile development environment. The study also demonstrated the importance and influence of macroculture on application of other social governance mechanisms.

Finally, I suggest that future research should look at the interactions amongst social governance mechanisms and their concurrency with especial investigation on macroculture content and its influence on other social governance mechanisms. I also recommended further research on the NG model comparing its application in different industries. I also suggest an extension of the current research as multi-method research on effective NG in distributed agile development projects.

References

Abrahamson E. & Fomburn C. (1992). Forging the iron cage: Interorganizational networks and the production of macroculture. *Journal of Management Studies,* 29(2): 175-194.

Abrahamsson P., Salo O., Ronkainen J and Warsta J. (2002). Agile software development methods - Review and analysis. *VTT Publications* 478:107.

Abrahamsson P., Conboy K. & Wang X. (2009). Lots done, more to do: the current state of agile systems development research. *European Journal of Information Systems,* 18(4): 281-284.

Ågerfalk P., Fitzgerald B. & Slaughter S. (2009). Flexible and Distributed Information Systems Development: State of the Art and Research Challenges. *Information Systems Research*, 20(3): 317–328.

Ambler W. S. (2009). Scaling agile software development through lean governance. *Proceeding of ICSE Workshop on Software Development Governance (SDG'09):* 1-2.

Anderson W. & Carney D. (2009). Distributed Project Governance Assessment (DPGA): Contextual, hands-on analysis for project governance across Sovereign boundaries. *Proceeding of ICSE Workshop on SDG'09:* 27-28, DIO. 10.1109/SDG.2009.5071333.

Anfara V. A., Brown K. M. & Mangione T. L. (2002). Qualitative analysis on stage: Making research process more public. *Educational Researcher*, 31(7), 28-38.

Apori-Nkansah L. (2008). Transitional justice in post conflict contexts: The case of Sierra Leone's dual accountability mechanisms. *Published doctoral dissertation*, Walden University, Minnesota, UMI, AAT 3291475.

Atwood J (2006). Anything but waterfall. [online] Retrieved from www.codinghorror.com/blog/archives/000694.html, [Accessed 27 Feb 2011].

Babar M. A. & Lescher C. (2014). Global software engineering: Identifying challenges is important and providing solutions is even better. *Information and Software Technology*, 56 (1): 1-5.

Bailey K. D. (1987). Methods of social Research. *Free press,* Collier Macmillan*,* 3rd ed., New York, NY.

Bannerman P.L. (2009). Software development governance: A meta-management perspective. *Proceeding of ICSE Workshop on SDG'09.* IEEE Computer Society, 3-8.

Baskerville R., Levine J., Pries-HejeJ., Ramesh B. & Slaughter S. (2001). How Internet Software Companies Negotiate Quality. *IEEE Computer*, 34(5): 51-57.

Bavani R. (2009). Critical Success Factors in Distributed Agile for Outsourced Product Development. *International Conference on Software Engineering, (CONSEG 09)*, Chennai, December 17-19, pp. 75-79.

Beck K. (1999b). Extreme Programming explained: Embrace change. *Reading, Mass., Addison-Wesley*.

Beck K. & Andres C. (2004). Extreme Programming Explained: Embrace Change. (2nd ed.). *Addison Wesley,* Boston.

Beck K., Beedle M., Bennekum A. V., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R. C., Mellor S., Schwaber K., Sutherland J. & Thomas D. (2001). Manifesto for Agile Software Development., [online] Retrieved from www.Agilemanifesto.org, [Accessed 24 April 2011].

Benbasat I., Goldstein D.K. & Mead M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, 11(3): 369-386.

Bennett A. & Elman C. (2010). Qualitative Research: Recent Development in Case Study Methods. *Annual Review of Political Science*, Palo Alto: Annual Review Publications.

Berg B. and Lune H. (2012). Qualitative Research Methods for the Social Sciences. *Pearson: London*.

Biffl S., Aurum A., Boehm B. and Erdogmus H. (2005). Value Based Software Engineering. *Springer: Heidelber*.

Boehm B. (2008). A Process Framework for System and Software Development Governance. *Proceedings of the 1st International Workshop on Software Development Governance (SDG'08): ACM,* pp. 1.

Boehm B. (2002). Get ready for the Agile methods, with care. *Computer*, 35(1): 64-69.

Bolton M. K., Malmrose R. and Ouchi W.G. (1994). The organization of innovation in the United States and Japan: Neoclassical and relational contracting. *Journal of Management Studies*, 31: 653-679.

Bourgeois L. J. & Eisenhardt K. M. (1988) "Politics of strategic decision making in high velocity environments: Toward a mid-range theory," *Academy of Management Journal*, 31(5): 737-770.

Boyatzis R.E. (1998). Transforming qualitative information: Thematic analysis and code development. *Thousand Oaks, London, & New Delhi: SAGE Publications.*

Boyce C. & Neale P. (2006). Conducting in-depth interviews: A guide for designing and conducting in-depth interviews for evaluation input. In: *Pathfinder International Tool Series: Monitoring and Evaluation.*

Brown C. V. (1999). Horizontal mechanisms under differing IS organization contexts' *MIS Quarterly*, 23(3): 421-454.

Burgelman R. A. (1983). A process model of internal corporate venturing in the diversified major firm. *Administrative Science Quarterly*, 28(2): 223-244.

Business Dictionary.com (2001). Retrieved from www.businessdictionary.com/definition/research-methodology.html, [Accessed 18 August 2015].
Cao L., Mohan K., Xu P. & Ramesh B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems*, 18(4): 332-343.

Cao L., Kannan M., Ramesh B & Sarkar S. (2013). Adapting funding processes for agile IT projects: as empirical investigation. *European Journal of Information Systems,* 22(2): 191-205.

Carmel E. (1999). Global software teams: collaborating across borders and time zones. *NJ: Prentice-Hall*.

Cassell C. & Symon G. (2004), "Essential guide to qualitative methods in organizational research" (323-333), *London, U.K: Sage*.

Cheng T., Jansen S. & Remmers M. (2009). Controlling and monitoring agile software development in three Dutch product software companies. *Proceeding of ICSE Workshop on SDG:* pp. 29-35.

Chulani S., Williams C. & Yaeli A. (2008). Software development governance and its concerns. *Proceedings of the 1st International workshop on SDG*, ACM: pp. 3-6.

Cockburn A. (2002). Agile software development: the business of innovation. *IEEE Computer Society,* 34(9): 120 – 127.

Collins M. & Thomas S. S. (2005). An exploration of problematic interviewee behaviours in qualitative research. *Western Journal of Nursing Research*, 27:188-199.

Conboy K. (2009). Agility from first principles: reconstructing the concept of agility in information systems development [Article]. *Information Systems Research*, 20: 329‐354.

Creswell J. W. (1998). Qualitative inquiry and research design: Choosing among five traditions. (1st ed.). *Thousand Oaks, CA: Sage*.

Creswell J. W. (2007). Qualitative Inquiry & Research Design: Choosing Among Five Approaches. (2nd ed.). *Thousand Oaks, CA: Sage Publications*.

Creswell J. W. (2009). Research design: qualitative, quantitative, and mixed methods approaches. (3rd ed.) *Thousand Oaks, CA: Sage Publications*.

Creswell J. W. & Miller D. (2000). Determining Validity in Qualitative Enquiry. *Theory into Practice,* 39(3): 124-130.

Cummings J. N. & Kiesler, S. (2007). Coordination costs and project outcomes in multi-university collaborations. *Science Direct*, 36(10): 1620-1634.

Charmaz K. (2006). Constructing grounded theory: A practical guide through qualitative analysis. *London: Sage.*

Crabtree B. F. & Miller W. L. (1992). Doing qualitative research. (2nd ed.). *Newbury Park, CA: Sage.*

Das T. & Teng B. (2001). Trust, control and risk in strategic alliances: an integrated framework. *Organization studies*, 22(2): 251-283.

Das T. & Teng B. (1996). Risk types and inter-firm alliance structures. *Journal of Management Studies,* 33(6): 827-843.

Denzin N. & Lincoln Y. (2003). Collecting and interpreting qualitative materials. *Thousand Oaks, CA: Sage.*

Denzin N. K. & Lincoln Y. (2005). Handbook of qualitative research. *Thousand Oaks, CA: Sage*.

Dey I. (1993). Qualitative data analysis: A user-friendly guide for social scientists. *London: Routledge.*

Dingsøyr T., Nerur S., Balijepally V. & Moe N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85 (6): 1213-1221.

Dingsøyr T. & Moe N. B. (2012). Research Challenges in Large-Scale Agile Software Development. *ACM SIGSOFT Software Engineering Notes*, 38(5): 38-39.

Dubé L. & Robey D. (2008). Surviving the paradoxes of virtual teamwork. *ISJ*, 19: 3-30. DOI:10.111/j.13652575.2008.00313.

Dubé L. & Paré G. (2003). Rigor in Information Systems Positivist Case Research: Current Practices, Trends and Recommendations. *MIS Quarterly* 27(4): 597-636.

Dubinsky Y., Chulani S. & Kruchten P. (2008). Software Development Governance. *Proceedings of the 1st International Workshop on Software Development Governance, ACM,* pp. 1059-1060.

Dubinsky Y. & Kruchten Ph. (2009). Software Development Governance (SDG). *31st International Conference on Software Engineering,* pp. 455-456.

Dubinsky Y. & Hazzan O. (2004). Development Teams. In Extreme Programming and Agile Processes in Software Engineering. *Springer, LNCS 3092*, pp. 157-165.

Dyba T. & Dingsoyr T. (2008). Empirical Studies of Agile Software Development: A Systematic Review. *Information and Software Technology*, 50: 833-859.

Eisenhardt K. (1989). Building Theory from Case Study Research. *Academy of Management Review,* 14(4): 532-550.

EngestÖm Y. (2000). Activity Theory as a Framework for Analysing and Redesigning Work. *Ergonomics*, 43(7):960-974.

Erbas C. & Erbas B. C. (2009). Software Development under Bounded Rationality and Opportunism. *ICSE Workshop on SDG:* 15-20.

Erdogmus H., Morisio M. & Torchiano M. (2005). On the effectiveness of the test-first approach to programming. *IEEE Transactions on Software Engineering*, 31(3): 226-237.

Faulkner R. R. & Anderson A. B. (1987). Short-term projects and emergent careers: Evidence from Hollywood. *American Journal of Sociology*, 92(4): 879-909.

Feller J., Finnegan P., Fitzgerald B. & Hayes J., (2008). From Peer Production to Productization: A Study of Socially Enabled Business Exchanges in Open Source Service Networks. *Information Systems Research*, 19(4): 475–493.

Fitzgerald B., Hartnett G. & Conboy K. (2006). Customising agile methods to software practices at intel Shannon. *European Journal of Information Systems*, 15(2):200-213.

Flyvbjerg B. (2006). Five misunderstandings about case-study research. *Qualitative Inquiry,* 72(2): 219-245.

Gallagher S. (1992). Hermeneutics and Education. Albany: *State University of New York Press*.

Geertz C. (1973). The interpretation of cultures. *New York, NY: Basic Books*.

Gentleman R. C., Carey V. J., Bates D. M., Bolstad B., Dettling M., Dudoit S., Ellis B., Gautier L., Ge Y., Gentry J., Hornik K., Hothorn T., Huber W., Iacus S., Irizarry R., Leisch F., Li C., Maechler M., Rossini A. J. , Sawitzki G., Smith C., Smyth G., Tierney L., Yang J. & Zhang J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology,* 5(10):1.

Ghauri P. (2004). Designing and Conducting Case Studies in International Business Research. *Edward Elgar Publishing Limited*.

Glaser B. G. & Strauss A.L. (1967). The discovery of grounded theory; strategies for qualitative research. *Chicago: Aldine pub.* Co.

Glesne C. & Peshkin A. (1992). Becoming Qualitative Researchers: An Introduction White plains. N.Y: *Longman*.

Golafshani (2003). Understanding reliability and validity in qualitative research. *The Qualitative Report*, 8 (4): 597-606.

Goulding C. (2002). Grounded Theory: A practical guide for management, business and market researchers. *London, U.K: Sage Publications*.

Granovetter M. (1992). Economic Institutions as Social Constructions: A framework for analysis. *ActaSociologica*, 35(1): 3-11.

Groenewald T. (2004). A phenomenological research design illustrated. *International Journal of Qualitative Methodsm,* 3(1): 14-19.

Grover V., Teng J. T. C. & Fiedler K. D. (2002). Investigating the Role of IT in Building Buyer-Supplier Relationships. *Journal of the Association for Information Systems*, 3(7): 217-245.

Guba E. G. & Lincoln Y. S. (1994). Competing paradigms in qualitative research. In N. K. Denzin & Y. S. Lincoln (Eds.). Handbook of Qualitative Research. (105-117) *Thousand Oaks, CA: Sage*.

Hajjdiab H., Taleb A. Sh. & Ali J. (2012). An Industrial Case Study for Scrum Adoption. *Journal of Software*, 7(1):237-242.

Hall D. & Hall I. (1996). Practice Social Research: project work in the Community. *Houndmills, Basingstoke: Macmillan*, pp.29.

Hall A.L. & Rist R. C. (1999). Integrating multiple research methods (or avoiding the precariousness of a one-legged stool). *Psychology & Marketing,* 16(4):291-304.

Hart C. (1998). Doing a literature review: Releasing the social science research imagination. *London, UK: Sage Publications*.

Hatch J.A. (2002). Doing Qualitative Research in educational settings. *Albany: State University of New York Press.*

Henderson-Sellers B. & Serour M. K. (2005). Creating a dual-agility method: the value of method engineering. *Journal of Database Management*, 4(14)1-16.

Highsmith J. & Cockburn A. (2001). Agile software development: the business of innovation. *IEEE Computer Society,* 34(1):120-127.

Hildenbrand T., Geisser M., Kude T., Bruch D. & Acker T. (2008). Agile Methodologies for Distributed Collaborative Development of Enterprise Applications. *International Conference on Complex, Intelligent and Software Intensive Systems (CISIS),* pp.540-545.

Hinds P. & Kiesler S. (2002). Distributed Work. *MIT Press*.

Hochstein A., Zarnekow R. & Brenner W. (2005). ITIL as common practice reference model for IT service management: formal assessment and implications for practice. *IEEE International conference on e-Technology (EEE'05), e-Commerce and e-Services,* pp. 704-710.

Hoegl M., Hans, W. & Gemuenden, G. (2004). Interteam Coordination, Project Commitment, and Teamwork in Multiteam R&D Projects: A Longitudinal Study. *Organization Science*, 15(1): 38-55.

Hossain E., Babar M. A. & Paik H. (2009). Using Scrum in GSD: A Systematic Literature Review. *Fourth IEEE International Conference on Global Software Engineering,* pp. 175-184.

Inayat I., Salim S. S., Marczak S., Daneva M., Shamshirband S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 51(1): 915-929.

ITGI. (2000). COBIT Executive Summary (3rd ed.). *IT Governance Institute*, *USA,* Retrieved from www.itgi.org, [Accessed 15 March 2011].

Janzen D. & Saiedian H. (2005). Test-driven development concepts, taxonomy, and future direction. *IEEE*, 38(9): 43-50.

Jeffries R., Anderson A. & Hendrickson C. (2001). Extreme Programming Installed. *Upper Saddle River, NJ, Addison-Wesley*.

Jensen M. C. (2001). A Theory of the Firm: Governance, Residual Claims, and Organizational Forms. *Cambridge, MA: Harvard University Press*, pp. 237.

Jiang J., Klein G. & Discenza R. (2002). Perception differences of software success: Provider and user views of system metrics. *Journal of Systems and Software*, 63(1):17-27.

Jones C., Hesterly W. S. & Borgatti S. P. (1997). A general theory of network governance: Exchange conditions and social mechanisms. *Academy of Management Journal*, 22(4): 911-945.

Jorgensen P. C. (2016). Software Testing: A Craftsman's Approach. *CRC Press Book.*

Jyothi V.E. & K.N. Rao. (2011). Effective Implementation of Agile Practices - Ingenious and Organized Theoretical Framework. *International Journal of Advanced Computer Science and Applications (IJACSA), New York, USA*, 2: 41-48.

Keng S. (2011). Cross-Disciplinary Models and Applications of Database Management: Advancing Approaches: Advancing Approaches. *Lincoln USA*, *IGI Global*.

Khther R. A. & Othaman M. (2013). COBIT framework as a guideline to effective governance in higher education. *International Journal of Information Technology Convergence and Services*, 3(1): 21-29.

Klein H. K. & Myers M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *Management Information Systems Quarterly*, 23(1):67-88.

Kollock P. (1994). The emergence of exchange structures: An experimental study of uncertainty, commitment, and trust. *The American Journal of Sociology,* 100(2): 313-345.

Korkala M. & Abrahamsson P. (2007). Communication in distributed agile development: A case study. *Software Engineering and Advanced Applications, 33rd EUROMICRO Conference:* pp.203-210.

Kvale S. (1992). Ten Standard Responses to Qualitative Research Interviews. *Thousand Oaks, CA: Sage.*

Kvale S. (2007). Doing interviews. *Thousand Oaks, CA: Sage*.

Kitzinger J. (1998). Qualitative research: Introducing focus groups. *BMJ*: 311-299.

Layder D (1998). Sociological practice: linking theory and social research. *London: Sage*.

Lee G. & Xia W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34: 87-114.

Lewis J. M. (2011). The future of network governance research: Strength in diversity and synthesis. *Public Administration*, 89(4): 1221-1234.

Lewis W. E. (2016). Software Testing and Continuous Quality Improvement*, Auerbach Publications, CRC Press.*

Linberg K.R. (1999). Software developer perceptions about software project failure: A case study. *Journal of Systems and Software*, 49(2-3): 177-192.

Lincoln Y. S. & Guba E. G. (1985). Naturalistic inquir. *Thousand Oaks, CA: Sage.*

Lincoln Y. (2001). Varieties of Validity: Quality in Qualitative Research. In Smart J. & Tierney W. (Eds.). Higher Education: Handbook of Theory and Research. New York: *Agathon Press. 16:* 25-72.

Lindvall M. & Rus I. (2000.) Process diversity in software development. *IEEE Software,* 17(4):14-71.

Lewis M. (2000). Exploring paradox: toward a more comprehensive guide. *Academy of Management Review*, 25(4): 760-776.

Mangalaraj G., Mahapatra R. & Nerur S. (2009). Acceptance of software process innovations - the case of extreme programming. *European Journal of Information Systems*, 18(4): 344-354.

Manhart P. & Schneider K. (2004). Breaking the ice for agile development of embedded software: an industry experience report. *Proceedings of the 26th International Conference on Software Engineering (ICSE), Edinburg: Scotland, UK,* pp. 378-386.

Maoz Z. (2002). Case study methodology in international studies: from storytelling to hypothesis testing. In Harvey F. P. & Brecher M. *Evaluating Methodology in International Studies*. University of Michigan Press, pp. 161-186.

Marschan-Piekkari R. & Welch C. (2004). Handbook of Qualitative Research Methods for International Business. UK: *Edward Elgar Publishing Limited:* 185-485.

Marchenko A. & Abrahamsson P. (2008). Scrum in a Multi-project Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges. *AGILE Conference,* IEEE, p. 15 -26.

Massa L. & Tucci C. L. (2013). Business Model Innovation. *The Oxford Handbook of Innovation Management*, PP. 420-441. DOI: 10.1093/oxfordhb/9780199694945.013.002.

Mattsson K. M., Azizyan G. & Magarian M. K. (2010). Classes of Distributed Agile Development Problems. *Agile Conference (AGILE),* P. 51-58.

Mckay J., Marshall P. & Smith L. (2003). Steps towards Effective IT Governance: Strategic IT Planning, Evaluation and Benefits Management. 7th *PACIS proceedings:* pp. 956-970.

McLoughlin I., Preece D. & Dawson P. (2000). Technology, Organizations and Innovation: Towards 'real virtuality?. (4th ed.) *Taylor & Francis.*

McReynolds C. J., Koch, L.C. & Rumrill, Jr. P. D. (2001). Qualitative research strategies in rehabilitation. *Work*, 16(1): 57-65.

Merriam S. B. (1998). Qualitative Research and Case Study Applications in Education. *United States of America: Jossey-Bass.*
Merriam S. B. (2002). Qualitative Research in practice: Examples for discussion and analysis. *New York, NY: Wiley, John & Sons.*

Merriam S. B. (2009). Qualitative Research: A guide to design and implementation: revised and expanded from Qualitative Research and Case Study applications in education. *San Francisco, CA: Jossey-Bass.*

Merriam S. B. & Tisdell E. J. (2015). Qualitative Research: A Guide to Design and Implementation. *CA: Jossey-Bass.*

Meso P. and Jain R. (2006). Agile Software development: Adaptive Systems Principles and Best Practices. *Information Systems Management*, 23: 19-30.

Meyer N. D. (2004). Systemic IS Governance: An Introduction. *Information Systems Management*, 21(4): 23-34.

Middleton P. (2001). Lean Software Development: Two Case Studies. *Software Quality Journal,* 9(4):241-252.

Miles M. B., Huberman A. M. and Saldaña J. (2013). Qualitative data analysis: A methods sourcebook. *Thousand Oaks*, CA: SAGE.

Miles M. B. and Huberman M. A. (1994). Qualitative Data Analysis: An Expanded Sourcebook. (2nd Eds.), USA: *Sage Publications*.

Moe N. B., Dingsøyr T. & Dybå T. (2009). Overcoming barriers to self-management in software teams. *IEEE Software*, 26(4):20-26.

Morgan G. & Smircich L. (1980). The case for qualitative research. *The Academy of Management Review*, 5(4):491-500.

Morrow S. L. (2005). Quality and trustworthiness in qualitative research in counselling psychology. *Journal of Counselling Psychology*, 52(2): 250-260.

Morse J. M. (1994). Designing funded qualitative research. In Denzin N. K. & Lincoln Y. S. (Eds.), Handbook of qualitative research. *Thousand Oaks, CA: Sage,* pp. 220-233.

Mumford E. Hirschheim R. A. Fitzgerald G. and WoodHarper T. (1985). Research Methods in Information Systems. *North Holland, Amsterdam*, pp. 201.

Münch J., Armbrust O., Kowalczyk M. & Soto M. (2012). Software Process Definition and Management. *Springer, Berlin, Heidelberg*.

Myers M. D. (1997). Qualitative Research in Information Systems. *MIS Quarterly* 21(2): 241-242.

Nachmias D. & Nachmias C. (1987). Research methods in the social sciences. (3rd ed.), *New York, NY: St. Martins Press.*

Nawrocki J., and Wojciechowski A. (2001). Experimental evaluation of pair programming. *12th European Software Control and Metrics Conference*, *ESCOM*, London: UK, pp. 269-276.

Nerur S., Mahapatra R. and Mangalaraj G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48: 73-78.

Nerur S. & Balijepally V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, 50(3): 79-83.

Neuman W. L. (1997). Social research methods: Qualitative and quantitative approaches. *Boston: Allyn and Bacon*.

Neuman W. L. (2003). Social research methods. *Boston, Allyn and Bacon.*

Orlikowski W.J. & Baroudi J.J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research,* 2(3): 1-28.

Passivaara M., Durasiewicz S. & Lassenius C. (2009). Using Scrum in Distributed Agile Development: A Multiple Case Study. In *4th IEEE International Conference on Global Software Engineering, I*CGSE'*09, Limerick, Ireland,* pp. 195-204.

Passivaara M., Durasiewicz S. & Lassenius C. (2008). Distributed agile development: Using Scrum in a globally distributed project: A case study. *IEEE International Conference on Global Software Engineering,* 13(6): 87-95.

Passivaara M. & Lassenius C. (2006). Could global software development benefit from Agile methods?. *International Conference on Global Software Engineering (ICGSE'06),* IEEE Computer Society, pp. 109-113.

Patton M. Q. (1987). How to use qualitative methods in evaluation. *Newbury Park, CA: Sage Publications.*

Patton M. Q. (1990). Qualitative Evaluation and Research Methods. *Thousand Oaks, CA: Sage Publications*.

Patton M. Q. (2002). Qualitative research and evaluation methods. (3rd ed.), *Thousand Oaks, CA: Sage Publications*.

Perry Ch. (1998). Processes of a case study methodology for postgraduate research in marketing. *European Journal of Marketing*, 32(9/10): 785-802.

Peterson R. R. (2001). Configurations and Coordination for Global Information Technology Governance: Complex Designs in a Transnational European Context. P*roceedings of 34th Hawaii International Conference on System Sciences, (HICSS),* pp. 10*.*

Poland B. (1995). Transcription Quality as an Aspect of Rigor in Qualitative Research. *Social Sciences, Qualitative Inquiry,* 1 (3): 290-310.

Polkinghorne D. E. (2005). Language and meaning: Data collection in qualitative research. *Journal of Counselling Psychology*, 52(2): 137-145.

Pooley R., Coady J., Schneider C., Linger H., Barry C. & Lang M. (2013). Information Systems Development, Reflections, Challenges and New Directions. *Proceedings of the 20th International Conference on Information Systems Development Cutting edge research on Information Systems, Springer NY*, pp. 2019-230.

Poppo L. & Zenger T. (2002). Do formal contracts and relational governance function as substitutes or complements?. *Strategy Management Journal*, 23(8):707-725.

Poppendieck M. & Poppendieck T. (2003). Lean Software Development: An Agile Toolkit. *Addison Wesley Professional,* pp. 28.

Powell W.W. (1990). Neither market nor hierarchy: Network forms of organizing. IN Staw B. & Cummings L. L. (Eds.). Research in organizational behaviour. *Greenwich, CT: JAI,* pp. 295-336.

Procaccino J. D. & Verner J.M. (2006). Software project managers and project success: An exploratory study. *Journal of Systems and Software,* 79(11): 1541-1551.

Punch K. F. (2005). Introduction to social research. *London, Thousand Oaks*, *New Delhi: Sage Publications.*

Pye G. & Warren M. J. (2006). Striking a balance between ethics and ICT 3 governance. *Australian Journal of Information Systems,* 13(2): 201-207.

Qumer A. (2007). Defining an integrated agile governance for large agile software development environments. *Agile processes in Software Engineering and Extreme Programming*, 4536: 157-160, DOI: 101007/978-3-54073101-6_23.

Raj S., Sepple J. & Willcocks L. (2013). IT governance: Spinning into control. *Outlook*, *The journal of high-performance business,* vol. 1.

Ramasubbu N & Balan R. K. (2008). Towards governance schemes for distributed software development projects. *Proceedings of the 1st international workshop on Software development governance*, *SDG'08*, *ACM New York, NY*, pp. 11-14.

Ramesh B., Cao L., Mohan K. & Xu P. (2006). Can distributed software development be Agile? *Communication of ACM,* 49(10): 40-52.

Rayhan S.H., Dhaka C. & Haque N. (2008). Incremental Adoption of Scrum for Successful Delivery of an IT Project in a Remote Setup. *Agile Conference, IEEE,* pp. 351-355.

Ribbers P., Peterson R. & Parker M. (2002). Designing information technology governance processes: diagnosing contemporary practices and competing theories. *Proceedings of the 35th Hawaii International Conference on System Sciences, (HICSS),* DOI: 10.1109/HICSS.2002.994351.

Ritchie J., Lewis J. & Gillian E. (2003). Designing and selecting samples. In Ritchie J. & Lewis J. (Eds.), Qualitative research practice: A guide for social science students and researchers. *Thousand Oaks, CA: Sage,* pp: 77-108.

Rosen S. M. (1994). Science, paradox, and the moebius principle. *Albany: State University of New York Press.*

Rubin H. & Rubin I. (2005). Qualitative interviewing: The art of hearing data. *Thousand Oaks: Sage Publications.*

Sangwan R., Bass M., Mullick N., Paulish D. J. & Kazmeier J. (2007). Global Software Development Handbook. *Auerbach Publications.*

Schwaber K & Beedle M. (2002). Agile Software Development with SCRUM. *Prentice-Hall*, Upper Saddle River, NJ.

Schwaber K. (2004). Agile Project Management with Scrum. *Microsoft Press*, Redmond, Washington.

Schwaber K., Laganza G. & D'Silva D. (2007). The Truth about Agile Processes: Frank Answers to Frequently Asked Questions. *Forrester Report*.

Shrivastava S. V. & Date H. (2010). Distributed Agile Software Development. Journal of Computer Science and Engineering, 1 (1): 10-17.

Shrivastava S. V. & Rathod U. (2015). Categorization of risk factors for distributed agile projects. *Information and Software Technology*, Elsevier 58: 373-387.

Silverman D. (2000). Doing Qualitative Research: A Practical Handbook. *Thousand Oaks: SAGE Publications*.

Singleton R. A. & Straits B. C. (2005). Approaches to Social Research. *New York, NY: Oxford University Press.*

Smith J. A., Flowers P. & Larkin M. (2013). Interpretative phenomenological analysis: Theory, method and research. *London: Sage Publications*.

Stainback S. B. & Stainback W. C. (1988). Understanding & Conducting Qualitative Research Council for Exceptional Children. *Social Science*.

Stake R. (1995). The Art of Case Research. *Thousand Oaks, CA: Sage Publications.*

Strauss L. A. & Corbin M. J. (1998). Basics of qualitative research: techniques and procedures for developing grounded theory. (2nd eds.), *SAGE Publication*.

Sundaramurthy C. & Lewis M. (2003). Control and collaboration: paradoxes of governance. *Academy of Management Review*, 28(3): 397-415.

Sureshchandra K. (2008). Adopting agile in distributed development. *IEEE International Conference on Global Software Engineering*, pp. 217-221.

Sutherland J., Viktorov A., Blount J. & Puntikov N. (2007). Distributed Scrum: Agile Project Management with Outsourced Development Teams. *Hawaii International Conference on Software Systems (HICSS'40)*, pp. 274.

Svensson H. & Höst M. (2005). Introducing an agile process in a software maintenance and evolution organization. *Proceedings of 9th European Conference of Maintenance and Reengineering*.

Tarr P., Williams C. & Hailpern B. (2008). Toward Governance of Emergent Processes and Adaptive Organizations. In *1st Workshop on SDG'08:*21-24, Germany.

Tharenou P. (2000). Ethical Issues and Conduct in the Practice of Research. *Book of Readings - Introductory Research Methods,* pp. 248-251.

Thomson E. & Vidgen R. (2013). Balancing the Paradox of Formal and Social Governance in Distributed Agile Development Projects. *Information Systems Development, Springer New York,* pp. 155-166.

Torfing J. (2005). Governance network theory: towards a second generation. *European Political Science,* 4: 305-315.

Trochim W.M.K. (2001). The research methods knowledge base. (2nd ed.), *United States of America: Atomic Dog Publishing.*

Turner D. (2010). Qualitative interview design: A practical guide for novice investigators. *The Qualitative Report*, 15(3): 754-760.

Unhelkar B. (2016). The Art of Agile Practice: A Composite Approach for Projects and Organizations. *Advanced & Emerging Communications Technologies,* CRC Press.

Usman M., Abbas Q., Akram B., Hussain J. (2016). Diminution of Issues and Challenges when Using Scrum in Global Software Engineering. *International Journal of Technology and Research*, 4 (2).

Vähäniitty J. & Rautiainen K. (2008). Towards a Conceptual Framework and Tool Support for Linking Long term Product and Business Planning with Agile Software Development. *Proceedings of the 1st Workshop on Software Development Governance*, ACM: 25-28.

Verschuren P. J. M. (2003). Case study as a research strategy: some ambiguities and opportunities. *International Journal of Social Research Methodology*, 6 (2): 121-139.

Vidgen R., Madsen S. & Kautz K. (2004). Mapping the Information System Development Process. *International Federation for Information Processing*, 141: 157-171.

Ward J., Daniel E. & Peppard J. (2008). Building Better Business Cases for IT Investments. *MISQE,* 7(1): 1-15.

Walsham G. (2006). Doing interpretive research European Journal of Information Systems. *Operational Research Society Ltd.*, 15: 320–330.

Walsham G. (1993). Interpreting Information Systems in Organizations. *Wiley, Chichester*.

Walsham G. (1995a). The emergence of interpretivism in IS research. *Information Systems Research*, 6(4):376-394.

Wang X., O'Conchuir E. & Vidgen R. (2008). A Paradoxical Perspective on Contradictions in Agile Software Development. *Proceedings of the 16th ECIS:* 9-11.

Webster J. & Watson R. T. (2002). Analysing the past to prepare for the future: writing a literature review. *MIS Quarterly*, 26(2): 13-23.

Weill P. & Ross J. W. (2004). IT governance on one page. *CISR WP* No. 49 and *MIT Sloan WP* No. 4516-04.

Weill P. & Woodham R. (2002). Don't Just Lead, Govern: Implementing Effective IT Governance. *CISRWP* No.326 and *MIT Sloan WP* No. 4237-02.

Wertz F. J. (2005). Phenomenological research methods for counselling psychology. *Journal of counselling psychology*, 52(2): 167-177.

Williams L., Kessler R. R., Cunningham W. & Jeffries R. (2000). Strengthening the case for pair programming. *IEEE Software*, 17:19-25.

Williamson O. E. (1994). Visible and invisible governance. *The American Economic Review,* 84(2): 323-326.

Winkler I. (2006). Network Governance between Individual and Collective Goals: Qualitative Evidence from Six Networks. *Journal of leadership and organizational studies*, 12(3): 119.

Yadav V., Ayda M., Nath D. & Sridhar V. (2007). Investigating an 'Agile-Rigid' Approach in Globally Distributed Requirements Analysis. *PACIS proceedings*.

Yaeli A. & Klinger T. (2009). Enacting responsibility assignment in software development environments. *Proceedings of the 1st International workshop on SDG*, ACM: 7-10.

Yin R. K. (1994). Case Study Research: Design and Methods. (2nd ed.), *Thousand Oaks, CA: Sage.*

Yin R. K. (2003). Case Study Research: Design and Methods. *SAGE Publications Inc*.

Yin R. K. (2009). Case study research. *Thousand Oaks, CA, Sage*.

Yin R. K. (2010). Qualitative research from start to finish. *New York: Guilford Press*.

Yin R. K. (2011). Qualitative research from start to finish. *New York, NY: The Guilford Press.*

Yin R. K. (2014). Case study research: Design and methods. 5th Ed. *Thousand Oaks, CA: Sage.*

Young C. & Terashima H. (2008). Agile conference How Did We Adapt Agile Processes to Our Distributed Development? Overcoming the Challenges of Building a Distributed Agile Organization. *In Proceedings of the Agile 2008 Conference*: 304-309.

Yu X. & Petter S. (2014). Understanding Agile Software Development Practices Using Shared Mental Models Theory. *Information and Software Technology,* 56 (8): 911-921.

Zalan T. & Lewis G. (2004). Writing About Methods in Qualitative Research: Towards a More Transparent Approach. In Marschan-Piekkari R. & Welch C. (Eds.), Handbook of qualitative research methods for international business. *Cheltenham: Edward Edgar*: 507-527.

Ziggers G. W., Gagalyuk T. & Hanf J. (2010). Network Governance at the Firm and Network Level: Goals, Routines and Social Mechanisms. *International Journal of Food System Dynamics,* 1 (4): 342-351.

Appendix A Case Study Protocol

1. Background

- identify previous research on the topic
- define the main research questions being addressed by this study
- identify any additional research questions that will be addressed

2. Design

- identify whether single-case or multiple-case and embedded or holistic designs will be used, and show the logical links between these and the research questions
- identify any sub-questions derived from each research question

3. Case Selection

- criteria for case selection
- case study procedures and roles
- procedures governing field procedures

5. Data Collection

- identify the data to be collected
- define a data collection plan
- define how the data will be stored

6. Analysis

- identify the criteria for interpreting case study findings
- identify which data elements are used to address which research question/sub question and how the data elements will be combined to answer the question
- consider the range of possible outcomes and identify alternative explanations of the outcomes, and identify any information that is needed to distinguish between these

7. Plan Validity (Lincoln 2001)

- credibility - show a causal relationship between outcomes and intervention/treatment (for explanatory or causal studies only)
- transferability - identify the domain to which study findings can be generalised. Tactics include using purposive sampling
- dependability - by triangulating the research findings from the different data sources. Tactics for ensuring this include using multiple sources of evidence, expert reviews of draft protocols and reports and using an audit trail
- confirmability - to show to what extent the data collected supports the research findings. Tactics for ensuring this include using different data sources

8. Reporting

- using a protocol template for case study planning
- identify target audience, relationship to larger studies (Yin, 2003)

9. Schedule

- give time estimates for all of the major steps: Planning, Data Collection, Data Analysis and Reporting.

10. Appendices

- divergences: update while conducting the study by noting any divergences from the above steps.

Appendix B Field Notes

Observation 1	Pseudonym: Date: 20/06/12
Observations: - Left the interview room for the daily Stand up - The Skype connection across three locations was very well arranged - The big TV screen where all team members were able to see each other all the time helped the transparency and coordination to remove the distance borders - The backlog and the posts on the wall to improve the transparency of activities - The project team settings were open as every team member was able to see others openly - The project team members were speaking freely and voiced their opinions - Facial expression shows that confidence and trust exist between team members - Good conversation	Comments to Questions: - Similarity in answers to questions about project goals and values - Lots of concerns about new evaluation system consistently coming up in different questions - Several questions answered with the use of phrases such as clarity of roles, high level of trust among the project team members, highly motivated - Agreed to follow-up interview, if any
Reflections: - Evidence of best practice - Evidence of strategies to prevent the project team members having fear of lock - Allowed respondent to digress from question in order for her to speak her heart. I will account for related themes under macroculture.	Reflections: - Some roles were described but not in the required detail - Need to clarify the roles' description and required details with the team leader

Observation 4	Pseudonym: Date: 19/11/12
Observations: - First Homemade Jam meeting - Sydney team were excited and were looking forward to start the first Homemade Jam meeting - The Melbourne team connected immediately - There was a delay in getting the China team to the meeting due to the internet connection error - Teams communicated very well and shared good knowledge - China and Sydney team interacted more compare to Melbourne team - Interesting discussion between team members about what worked really well during the last project phase and what they learned to implement it in the next phase - The communication between teams were clear and facial expressions and confidence in voice tones showed clear understanding of the subject and in communication - The interactions and the knowledge transferred also cleared some context related issues among the teams that could improve the coordination among them during the next iterations - Good knowledge transfer and sharing	Comments to Questions: - The more interaction between Sydney and China team was explained by the IM as Sydney and China teams share similar tasks as front developers but Melbourne team are more mainframe developers - Similarity in answers to questions about shared project goals - Several comments in answering the questions about the importance of having a clear understanding about the context among team across borders - Agreed to follow-up interview, if any
Reflections: - Evidence of best practice - Evidence of sharing goals and strategies	Reflections: - Need to clarify some of the technical terms used - More details required about the context sharing among teams across borders

- Frequent interactions among teams during the meeting allowed them to clear context issues and improve the coordination across borders	

Observation 3	Pseudonym: Date: 16/11/12
Observations: - All the senior managers of the project attended the Showcase meeting - Few core team members from Sydney team were present - The discussion was more related to the strategies - Very little detailed discussed about the technical and lower level developing phase - Lots of interactions between the product owner, project manager and program manager - IM and Sydney BA representing the Sydney team - Melbourne and China team joined the meeting through Skype on time - The internet connection was very clear - Not much participation from China developers but their BA - Good conversation among senior management of the project	Comments to Questions: - Similarity in answers to questions about status maximisation - restricted access - Several questions were answered with the use of team members with 'similar interests', limited access and similar status tendency to interact more frequently - Agreed to follow-up interview, if any
Reflections: - Evidence of best practice - Evidence of trust and confidence senior management have in the project teams	Reflections: - More clarification required about the interaction between team members with similar interests - Need to get more details when China team has restricted access to the project

Appendix C Letter of Invitation to Potential Participants

To Whom It May Concern:

Allow me to introduce myself. My name is Esmeralda Thomson and I am a student pursuing a Ph.D. in Adoptive Governance Mechanisms in Distributed Agile Development Projects, at the University of New South Wales (UNSW). I have written this letter as an invitation for you to participate in a doctoral research study exploring how to use social governance mechanisms effectively in distributed agile development projects to enhance coordination and safeguarding exchanges among the project teams across locations. The research aims to provide best practices for an effective governance of distributed agile development projects. Therefore, this research explores how social governance mechanisms applied in distributed agile development projects and how they coordinate and protect project interactions across geography.

This study will fulfil my dissertation requirement for this program. This research will be supervised by my supervisors A/Prof. Kieran Conboy and Dr. Ken Stevens. You were selected as potential participant due to your previous experience and you role in the project. Your participation in this research will be very much appreciated because it will ultimately add to the body of knowledge available to organisations.

This study is a qualitative research, which involves a tape recorded interviews with participants using a pseudonym, which may last approximately 90 minutes. There may be a follow-up interview by telephone for any clarification. The interviews will be held at the project site or any other mutually agreeable location at a time convenient to you. In order to assure the validity of this research, you may be given draft conclusions to clarify. I will conduct all the interviews and follow-up sessions to keep your identity confidential.

Your participation is purely voluntary and you may withdraw, without any penalty, at any point, from this study. At all times, information from the interviews will be kept confidential and your identity protected.

I would be interested in scheduling our interview sometime in the first week of June, 2012. Your participation is much appreciated and I will follow up via phone or email to confirm your interest. Please find attached a consent form for you to sign and return to me, if you decide to participate in this study. You can also respond to this letter by emailing me at albert.essandoh@gmail.com

Thank you for considering my request.

Yours Sincerely,
Esmeralda Thomson

Appendix D Interview Protocol

The following is the document that was used in the interviews:

Introductory remarks (notes for interviewer)

Introduce myself
Describe research context and goals in brief
Signing interview recording consent
Background information on interviewee:

- Date
- Name
- Contact details
- Team location
- Title
- Primary functions of the current role involved
- Education background
- Years of experience (previous/current)
- Previous work experience

Standard Questions

RA) Restricted Access – reducing participant for certain types of exchange

RA1) How strong do you consider the ties between project team members? Why?

RA2) Do project team members find it easier to interact with the similar status level members across locations? How?

RA3) How about trust among project team members across location? How do Sydney, China and Melbourne team members see each other as reliable and trustworthy? Examples

RA4) Through your interactions with stakeholders, do you have a relational contract? Give me an example

RA5) How clear are the project roles for project team members across geography? Give me an example

RA6) Have you faced any challenges through exchanges among project team members? What did you do about it? Give me an example

M) Macroculture - shared common norms, values, goals and assumptions

M1) Do project team members share a sense of belonging across geography? Give me an example

M2) How about values that were shared between all project members across geography? Give me an example

M3) Do project team members accept a similar way of working that is shared between all locations? Give me an example

M4) Is there a culture of all project team members commonly agreeing on the project goals?

M5) Did you share any communicating protocol, for example common approach and solutions to a certain situation in all locations?

M6) How about any shared assumptions across locations? i.e. the important tasks?

C) Collective Sanctions

C1) Have you experienced any issues about project team members' unacceptable behaviours across geography? What did you do about it?

C2) If there is a team member who is not very active and involved in the team, what would be the strategy to face this?

C3) Do you have a rewarding strategy for the team members who behave properly? How?

C4) How do you think these strategies impact on safeguarding the interactions among the team members across locations?

C5) How do they impact on coordination across locations? Give me an example

C6) Have you become aware of any challenges in this regard? What did you do about it? Give me an example.

R) Reputation

R1) Do the project team meet the social expectation? Give me an example

R2) Do you think the team members consider it important to fulfil their obligations towards other team members in order to keep their positive reputation? Give me an example

R3) Do the team members consider it important to be considered reliable and trustworthy by other team members? Give me an example

R4) Do you think the team members consider it important that other team members consider them professionally competent? How?

R5) How strong do you think having a positive reputation is among the project team across locations?

R6) Do you think reputation between project team members across locations is based on trust? Why?

R7) Have you become aware of any reputation challenges? What did you do about it? Give me an example

S) Supplementary questions

S1) Have you had any cultural related challenges?

S2) What did you do about it? Give me an example

S3) What are the consequences of team members' unacceptable behaviours?

S4) Does your strategy about not very involved team members work?

S5) If it doesn't work, what would you do next?

S6) What is the use of the story wall?

S7) How often do you do the continuous testing and who does it?

S8) How often do you have iteration planning meetings and retrospectives? Who are the participants?

S9) What do you mean by 'success slider'?

F) Follow up questions

F1) Can you give me an example where you shared a common approach or solutions to a certain situation in all locations?

F2) Can you give me an example where team members had consistent interests?

F3) Can you tell me about the communication tools shared between teams across borders?What are the key practices of Ocean Group agile approach?

F4) What is KRA?

F5) How do you ensure the information and knowledge asymmetries across locations?

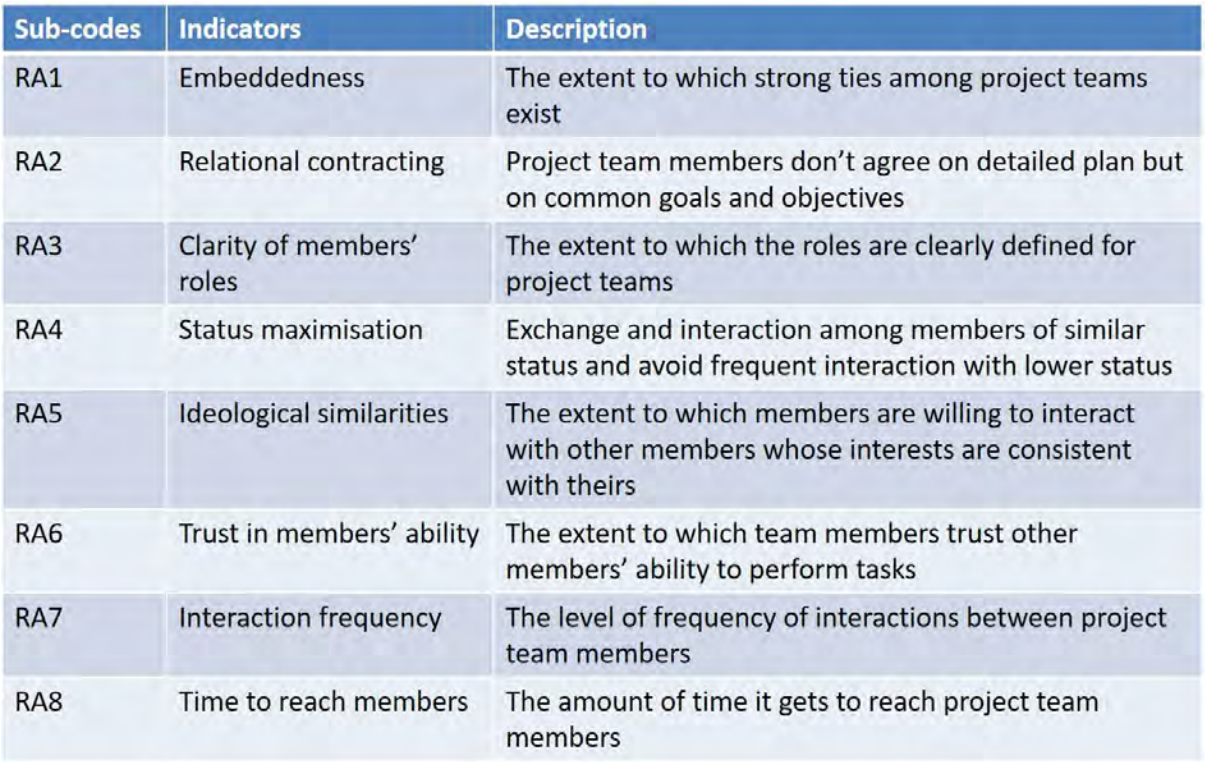
I) Interview Close

I1) Ask if interviewee has any further questions

I2) Ask if interviewee has anything to add

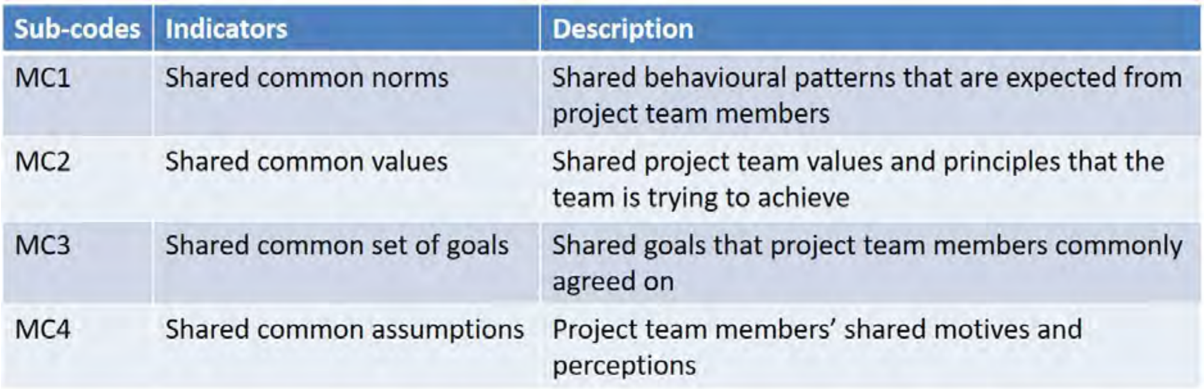
Appendix E Code Book 2 Tables

Restricted Access Coding – Level 2 (Code book 2)



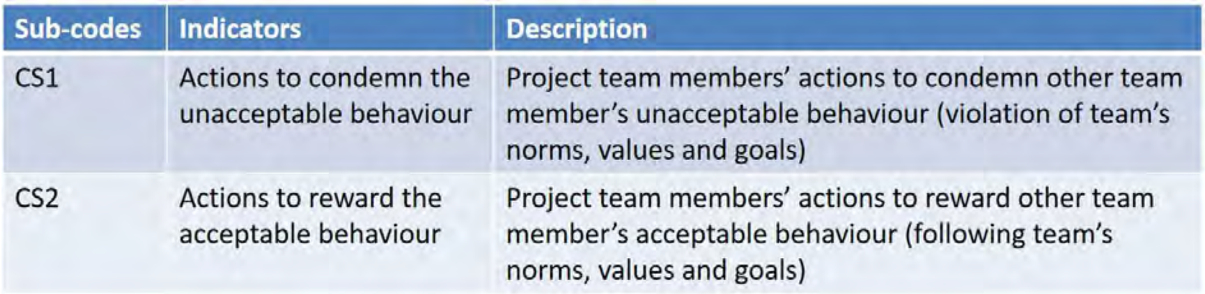
Sub-codes	Indicators	Description
RA1	Embeddedness	The extent to which strong ties among project teams exist
RA2	Relational contracting	Project team members don't agree on detailed plan but on common goals and objectives
RA3	Clarity of members' roles	The extent to which the roles are clearly defined for project teams
RA4	Status maximisation	Exchange and interaction among members of similar status and avoid frequent interaction with lower status
RA5	Ideological similarities	The extent to which members are willing to interact with other members whose interests are consistent with theirs
RA6	Trust in members' ability	The extent to which team members trust other members' ability to perform tasks
RA7	Interaction frequency	The level of frequency of interactions between project team members
RA8	Time to reach members	The amount of time it gets to reach project team members

Macroculture Coding – Level 2 (Code book 2)



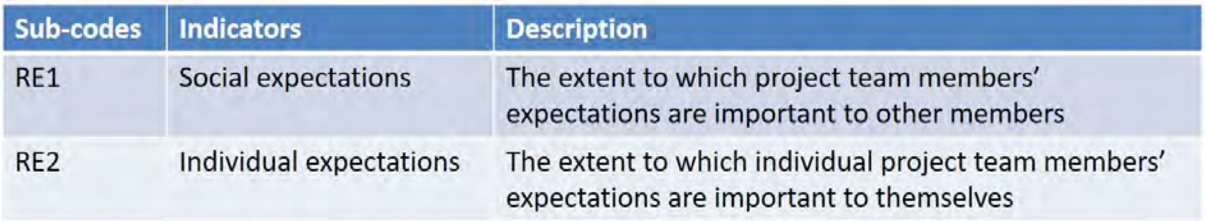
Sub-codes	Indicators	Description
MC1	Shared common norms	Shared behavioural patterns that are expected from project team members
MC2	Shared common values	Shared project team values and principles that the team is trying to achieve
MC3	Shared common set of goals	Shared goals that project team members commonly agreed on
MC4	Shared common assumptions	Project team members' shared motives and perceptions

Collective Sanctions Coding – Level 2 (Code book 2)



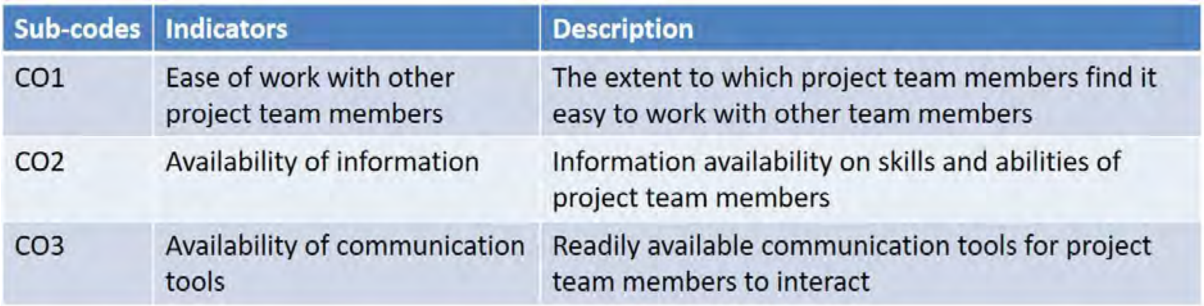
Sub-codes	Indicators	Description
CS1	Actions to condemn the unacceptable behaviour	Project team members' actions to condemn other team member's unacceptable behaviour (violation of team's norms, values and goals)
CS2	Actions to reward the acceptable behaviour	Project team members' actions to reward other team member's acceptable behaviour (following team's norms, values and goals)

Reputation Coding – Level 2 (Code book 2)



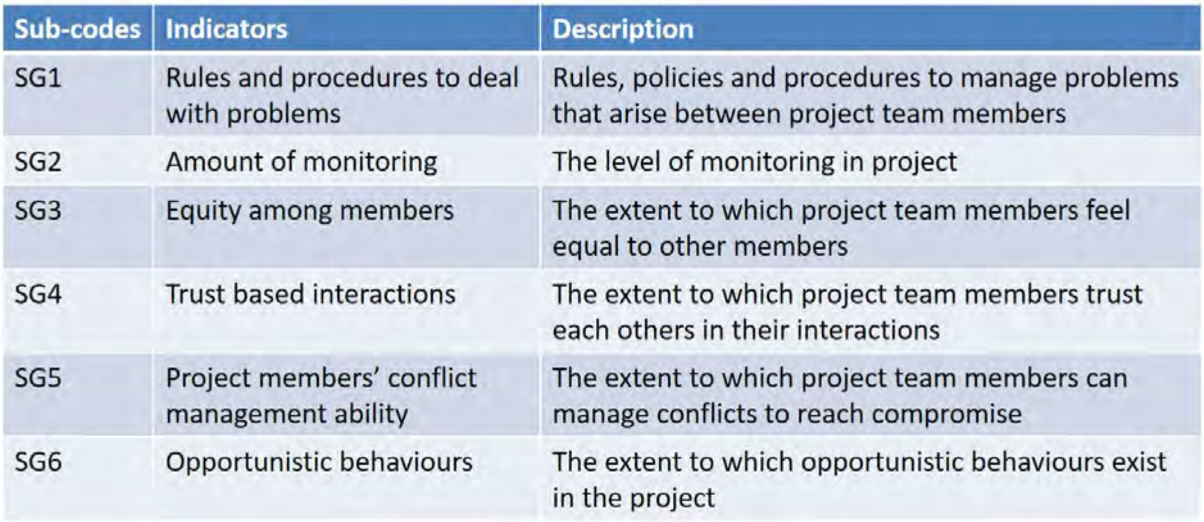
Sub-codes	Indicators	Description
RE1	Social expectations	The extent to which project team members' expectations are important to other members
RE2	Individual expectations	The extent to which individual project team members' expectations are important to themselves

Coordination Coding – Level 2 (Code book 2)



Sub-codes	Indicators	Description
CO1	Ease of work with other project team members	The extent to which project team members find it easy to work with other team members
CO2	Availability of information	Information availability on skills and abilities of project team members
CO3	Availability of communication tools	Readily available communication tools for project team members to interact

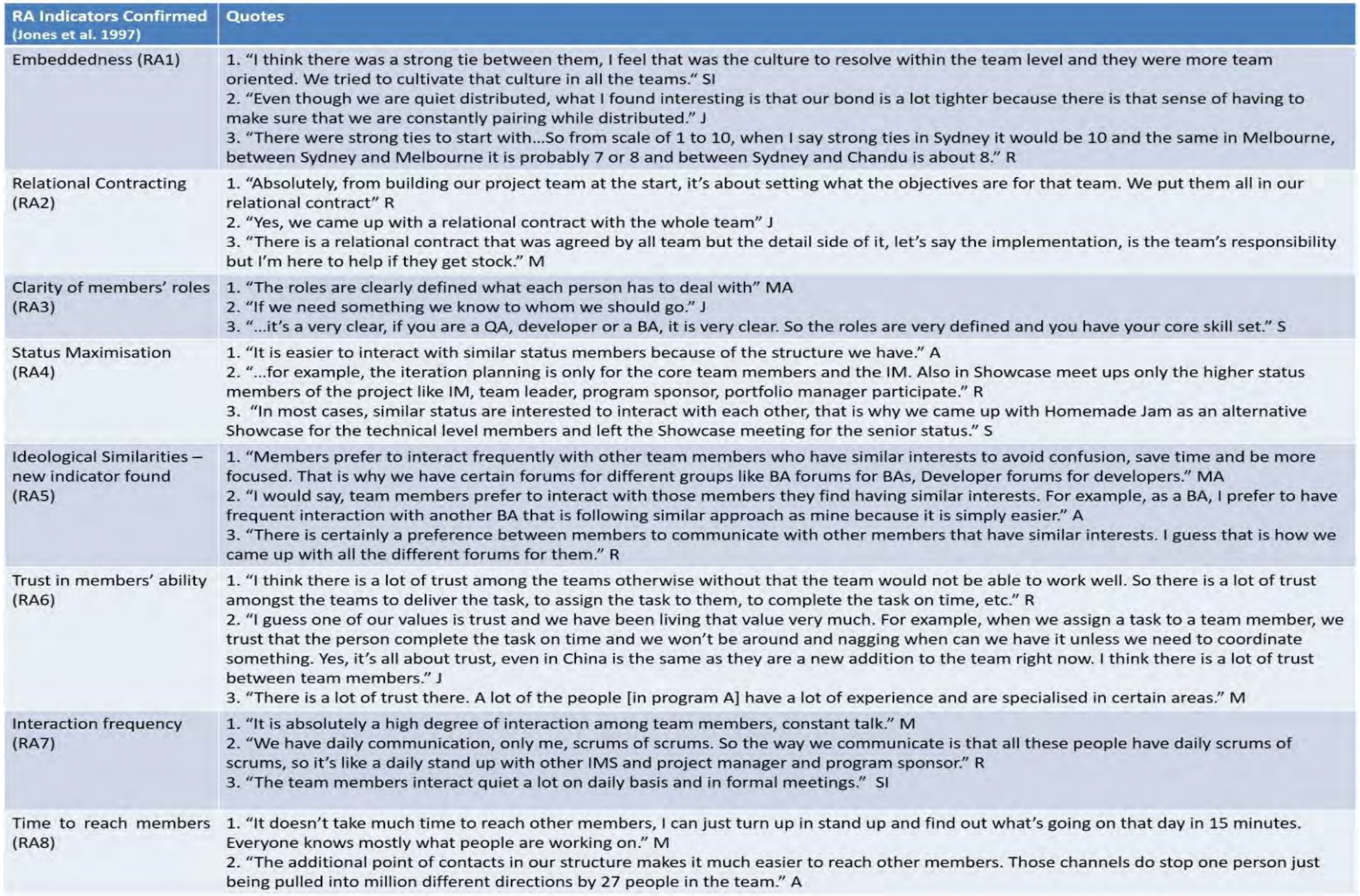
Safeguarding Coding – Level 2 (Code book 2)



Sub-codes	Indicators	Description
SG1	Rules and procedures to deal with problems	Rules, policies and procedures to manage problems that arise between project team members
SG2	Amount of monitoring	The level of monitoring in project
SG3	Equity among members	The extent to which project team members feel equal to other members
SG4	Trust based interactions	The extent to which project team members trust each others in their interactions
SG5	Project members' conflict management ability	The extent to which project team members can manage conflicts to reach compromise
SG6	Opportunistic behaviours	The extent to which opportunistic behaviours exist in the project

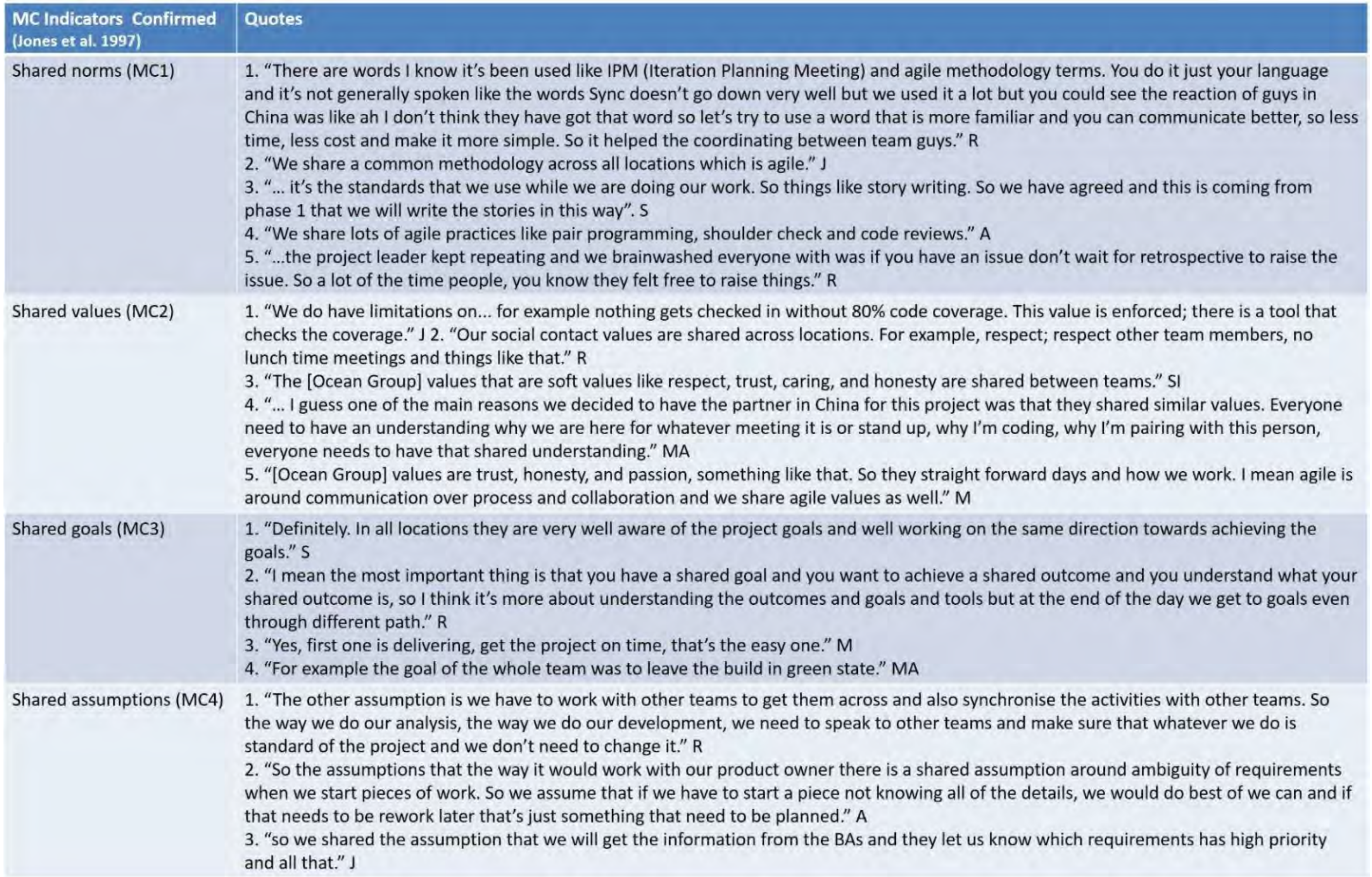
Appendix F Constructs Codes and Quotes – An excerpt of Code Book 2

RA Codes and the Quotes



RA Indicators Confirmed (Jones et al. 1997)	Quotes
Embeddedness (RA1)	1. “I think there was a strong tie between them, I feel that was the culture to resolve within the team level and they were more team oriented. We tried to cultivate that culture in all the teams.” SI 2. “Even though we are quiet distributed, what I found interesting is that our bond is a lot tighter because there is that sense of having to make sure that we are constantly pairing while distributed.” J 3. “There were strong ties to start with…So from scale of 1 to 10, when I say strong ties in Sydney it would be 10 and the same in Melbourne, between Sydney and Melbourne it is probably 7 or 8 and between Sydney and Chandu is about 8.” R
Relational Contracting (RA2)	1. “Absolutely, from building our project team at the start, it’s about setting what the objectives are for that team. We put them all in our relational contract” R 2. “Yes, we came up with a relational contract with the whole team” J 3. “There is a relational contract that was agreed by all team but the detail side of it, let’s say the implementation, is the team’s responsibility but I’m here to help if they get stock.” M
Clarity of members’ roles (RA3)	1. “The roles are clearly defined what each person has to deal with” MA 2. “If we need something we know to whom we should go.” J 3. “…it’s a very clear, if you are a QA, developer or a BA, it is very clear. So the roles are very defined and you have your core skill set.” S
Status Maximisation (RA4)	1. “It is easier to interact with similar status members because of the structure we have.” A 2. “…for example, the iteration planning is only for the core team members and the IM. Also in Showcase meet ups only the higher status members of the project like IM, team leader, program sponsor, portfolio manager participate.” R 3. “In most cases, similar status are interested to interact with each other, that is why we came up with Homemade Jam as an alternative Showcase for the technical level members and left the Showcase meeting for the senior status.” S
Ideological Similarities – new indicator found (RA5)	1. “Members prefer to interact frequently with other team members who have similar interests to avoid confusion, save time and be more focused. That is why we have certain forums for different groups like BA forums for BAs, Developer forums for developers.” MA 2. “I would say, team members prefer to interact with those members they find having similar interests. For example, as a BA, I prefer to have frequent interaction with another BA that is following similar approach as mine because it is simply easier.” A 3. “There is certainly a preference between members to communicate with other members that have similar interests. I guess that is how we came up with all the different forums for them.” R
Trust in members’ ability (RA6)	1. “I think there is a lot of trust among the teams otherwise without that the team would not be able to work well. So there is a lot of trust amongst the teams to deliver the task, to assign the task to them, to complete the task on time, etc.” R 2. “I guess one of our values is trust and we have been living that value very much. For example, when we assign a task to a team member, we trust that the person complete the task on time and we won’t be around and nagging when can we have it unless we need to coordinate something. Yes, it’s all about trust, even in China is the same as they are a new addition to the team right now. I think there is a lot of trust between team members.” J 3. “There is a lot of trust there. A lot of the people [in program A] have a lot of experience and are specialised in certain areas.” M
Interaction frequency (RA7)	1. “It is absolutely a high degree of interaction among team members, constant talk.” M 2. “We have daily communication, only me, scrums of scrums. So the way we communicate is that all these people have daily scrums of scrums, so it’s like a daily stand up with other IMS and project manager and program sponsor.” R 3. “The team members interact quiet a lot on daily basis and in formal meetings.” SI
Time to reach members (RA8)	1. “It doesn’t take much time to reach other members, I can just turn up in stand up and find out what’s going on that day in 15 minutes. Everyone knows mostly what people are working on.” M 2. “The additional point of contacts in our structure makes it much easier to reach other members. Those channels do stop one person just being pulled into million different directions by 27 people in the team.” A

MC Codes and Quotes



MC Indicators Confirmed (Jones et al. 1997)	Quotes
Shared norms (MC1)	1. “There are words I know it’s been used like IPM (Iteration Planning Meeting) and agile methodology terms. You do it just your language and it’s not generally spoken like the words Sync doesn’t go down very well but we used it a lot but you could see the reaction of guys in China was like ah I don’t think they have got that word so let’s try to use a word that is more familiar and you can communicate better, so less time, less cost and make it more simple. So it helped the coordinating between team guys.” R 2. “We share a common methodology across all locations which is agile.” J 3. “… it’s the standards that we use while we are doing our work. So things like story writing. So we have agreed and this is coming from phase 1 that we will write the stories in this way”. S 4. “We share lots of agile practices like pair programming, shoulder check and code reviews.” A 5. “…the project leader kept repeating and we brainwashed everyone with was if you have an issue don’t wait for retrospective to raise the issue. So a lot of the time people, you know they felt free to raise things.” R
Shared values (MC2)	1. “We do have limitations on… for example nothing gets checked in without 80% code coverage. This value is enforced; there is a tool that checks the coverage.” J 2. “Our social contact values are shared across locations. For example, respect; respect other team members, no lunch time meetings and things like that.” R 3. “The [Ocean Group] values that are soft values like respect, trust, caring, and honesty are shared between teams.” SI 4. “… I guess one of the main reasons we decided to have the partner in China for this project was that they shared similar values. Everyone need to have an understanding why we are here for whatever meeting it is or stand up, why I’m coding, why I’m pairing with this person, everyone needs to have that shared understanding.” MA 5. “[Ocean Group] values are trust, honesty, and passion, something like that. So they straight forward days and how we work. I mean agile is around communication over process and collaboration and we share agile values as well.” M
Shared goals (MC3)	1. “Definitely. In all locations they are very well aware of the project goals and well working on the same direction towards achieving the goals.” S 2. “I mean the most important thing is that you have a shared goal and you want to achieve a shared outcome and you understand what your shared outcome is, so I think it’s more about understanding the outcomes and goals and tools but at the end of the day we get to goals even through different path.” R 3. “Yes, first one is delivering, get the project on time, that’s the easy one.” M 4. “For example the goal of the whole team was to leave the build in green state.” MA
Shared assumptions (MC4)	1. “The other assumption is we have to work with other teams to get them across and also synchronise the activities with other teams. So the way we do our analysis, the way we do our development, we need to speak to other teams and make sure that whatever we do is standard of the project and we don’t need to change it.” R 2. “So the assumptions that the way it would work with our product owner there is a shared assumption around ambiguity of requirements when we start pieces of work. So we assume that if we have to start a piece not knowing all of the details, we would do best of we can and if that needs to be rework later that’s just something that need to be planned.” A 3. “so we shared the assumption that we will get the information from the BAs and they let us know which requirements has high priority and all that.” J

CS Codes and Quotes

CS Indicators Confirmed (Jones et al. 1997)	Quotes
Actions to condemn the unacceptable behaviour (CS1)	1. “The extreme and this actually happened at one point, is the removed from the project.” S 2. “Encouraging them gently. So if I’m sitting in a meeting and I know this person knows what we are talking about but not saying anything, I might just go: hey what do you think? Do you think it’s all right?” R 3. “They talk to the person if the person is not performing well, if it does not work escalate it to the HR leader and because are on bounce system here it could have financial consequences, because you have to have a minimum successful level in the annual performance review. If you do not get to your successful level you do not get your bounce and no salary increases or promotions. And if you are overly performing you will get the bonuses, promotion and salary increase.” MA 4. “Beside the financial consequences, there are also non-financial consequences. For example, you might be spoken to and you know, no one to get negative feedback from the leader, but we try to resolve thing at the first level before escalating it. Agile is a team that they try to pull each other up especially with the high performance team does not like the not contributing members and don’t have much tolerance for that.” SI 5. “Consequences are more about gradual escalation, so in relation to how much feedback they get and the directness of the feedback.” M
Actions to reward the acceptable behaviour (CS2)	1. “We have a Thank You system that basically broadcast a ‘thank you’ message for a particular team member to all members. Also, through positive feedback and at the showcase it happens more frequently when we show to the business this is what the team has done, so a lot of acknowledgement and even constructive feedbacks and suggestion for improvements.” J 2. “…awarding people for when they do things and recognising that they’ve done things well, that they shared something that helped us.” R 3. “I think the other one was you know you really want to make it safe to fail, you want people to express or make it clear when they’ve failed or make it clear that that’s the wrong behaviour . So as much as it’s model to right behaviour it’s also model to wrong behaviour and identify such.” A 4. “We reward the well performed members financially as well through our bonus system”. MA

RE Codes and Quotes

RE Indicators Confirmed (Jones et al. 1997)	Quotes
Social expectations (RE1)	1. “I guess agile supports that flexibility but you would say you are going to get it done by tomorrow and it will take a week that will affect your brand and how people think of you and you keep doing it and saying oh I will get it done by tomorrow and every time it will take a week time.” M 2. “Because other project team members across all locations have high visibility of your actions and your activities are very transparent to others, you don’t have a chance to hide your activities. This motivates team members to present positive reputation towards other members as much as possible and fulfil expectations of other team members. So, having that visibility and transparency amongst all of your team members, as developer I used to have more motivation to try.” J 3. “I think there is a big sense of team and team members do think it’s very important not to let other team members down. Not only for reputation of the company and partners that are involved but I think it’s keeping the team together … it’s a personal obligation that everybody feels.” R
Individual expectations (RE2)	1. “Well I think probably the importance of individual reputation, when we do things, really points out. So when have pairing across countries, find the first people to pick it up was a tech lead and everyone followed that person because everyone looked up to that person as one the stronger people in the team.” SI 2. 1. “There is an expectation that everybody does that because it helps the team get along. So, there is an expectation that people will perform to their role but there is also an understanding that there is different skill levels within the team.” S

Appendix G Participant Information Statement and Consent Form

Approval Number: 116010

THE UNIVERSITY OF NEW SOUTH WALES

Adoptive Governance in Distributed Agile Development Projects

Participant selection and purpose of study

You, the research participant, are invited to participate in a study of governance for distributed agile development projects. I, Esmeralda Thomson hope to learn and evaluate the governance mechanisms used to manage distributed agile development project teams. You were selected as a possible participant in this study because of your close involvement in distributed agile development projects.

Description of study and risks

If you decide to participate, I will be conducting an interview, in which I present a few questions and invite you to answer to the best of your knowledge. This process should be expected to take around 30 minutes to 1 hour to complete (depending on the interview).

Benefits you can expect to gain from your involvement may include learning more about the ways in which different mechanisms for governance of distributed development projects are evaluated, as well as contributing to the best practices to enhance the governance of distributed agile development projects.

I cannot and do not guarantee or promise that you will receive any benefits from this study.

Please note that your participation may be recorded via a voice recording device and stored onto a tape(s). This information will be used to help with the note-taking process. This will be notified before the interview, and will be undertaken at your discretion. If you do not wish to be recorded onto audio tape, the interview will proceed and only hand written/computer-written notes will be made.

Confidentiality and disclosure of information

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission, except as required by law. If you give me your permission by signing this document, I plan to publish the results in certain relevant information systems academic research journals. In any publication, information will be provided in such a way that you cannot be identified.

Recompense to participants

Complaints may be directed to:

Ethics Secretariat,
The University of New South Wales
Sydney 2052 AUSTRALIA

Phone: +61 2 9385 4234
Fax: +61 2 9385 6648
Email: ethics.sec@unsw.edu.au).

Any complaint you make will be investigated promptly and you will be informed out the outcome.

Feedback to participants

If you wish, I can provide a summary of research upon completion of the project. This will be in the form of feedback, some statistical findings and/or summaries as appropriate, and I will contact you by email (please leave your email address at the bottom of your consent form if you wish to be contacted with these results later).

Your consent

Your decision whether or not to participate will not prejudice your future relations with the University of New South Wales. If you decide to participate, you are free to withdraw your consent and to discontinue participation at any time without prejudice.

If you have any questions, please feel free to ask me. If you have any additional questions later, either myself, Esmeralda Thomson (ph: +61 424 044 979, email: Esmeralda@student.unsw.edu.au) or my research supervisors, A/Prof. Kieran Conboy (ph: +61 2 9385 6924, email: k.conboy@unsw.edu.au) and Dr. Ken Stevens (ph: +61 2 9385 4242, email: k.stevens@unsw.edu.au) will be happy to answer them.

You will be given a copy of this form to keep.

Appendix H Letter of Appreciation

Dear project member,

Thank you for taking the time to participate in this case study about how social governance mechanisms can be adopted effectively in distributed agile development projects. I appreciate the time you spent responding to all my questions and providing additional information, when needed. Your feedback was very helpful and enabled me to understand the practices you use in the projects.

I have completed the study and made some recommendations for distributed development projects, and I hope it will be beneficial to your organisation, as well.

When the research report is formally approved, I will send your organisation a copy for your perusal.

Thank you

Esmeralda Thomson

Appendix I The Case Project Artefacts

Document Type	Document Name	Where listed
Project artefacts	The project structure chart	Appendix J
	Ocean Group values statement	Appendix H (item 1)
	River values statement	Appendix H (item 2)
	Lake project relational contract document	Appendix H (item 3)
	IT HR structure chart	Figure 4.2
	Program structure chart	Figure 4.3
	Project status report	Appendix H (item 4)
	Project team activity report	Appendix H (item 5)
	Retrospective plan	Appendix H (item 6)
	The team meeting minutes	Appendix H (item 7)
	Project Performance report	Appendix H (item 8)
Website data	The case company web site	Information obtained used in section 4.2.2, 4.2.3 and 4.2.4
Interviews	Semi-structure interviews	Appendix D

Item 1: Ocean Group Values

The Ocean Group staff value honesty, trust and transparency, reject discrimination and inequality and promote diversity in all its forms. The Ocean Group values listed below:

- Trust – keeping promises
- Honesty – being genuine and ethical
- Courage – taking responsibility of actions, being up front to voice mistakes and failures
- Caring – listening carefully to others and working towards shared goals
- Fairness – treating people equally
- Respect – treating people with dignity

Item 2: River Values

- Do the right thing
- Attitude, aptitude and integrity
- Service to others and society over self
- Solidarity over charity
- Serve holistic goals over achieving targets
- Personal and organisational transparency
- Curiosity, creativity and passion
- Fail fast and publish our mistakes
- Intolerant of intolerance
- No jerks
- High values alignment, loosely coupled

The Ocean Group and River staff shared agile values as well:

Agile Values (Beck K. *et al.* 2001)

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Item 3: Lake Project Relational Contract

- Keep on time to meetings
- Code peer review before SYST – Protect
- Be mindful of people's time when booking meetings
- Ensure meetings have relevant outcomes and people
- Face to face when possible
- Keep good time
- Every review is valid
- Freedom to withdraw if unnecessary
- If it needs a brain it needs a pair
- No check-ins on red build
- Do not leave broken build overnight
- Frequent tech huddles
- Course correction not course perfection
- Keep JIRA up to date

Item 4: Project Status Report

I viewed the report on 11 July 2012. The actual report cannot be provided through commercial in confidence however the status report was for June 2012 explaining a history of the project, including description and milestones, which were useful in terms of tracking progress, evaluation and review. It also included a budget report with respect to planned expenditure, risk management report that specified any changes to the major risks identified and the strategies to manage them. The status report also contained issues report including areas of concern and specific problems discussed.

Item 5: Project team activity report

I viewed the report on 15 August 2012. The actual report cannot be provided through commercial in confidence however the report was for July 2012 containing details of the activities and progress of the team members. The report is about the team's performance and an insight on who is doing what, when, and how.

Item 6: Retrospective Plan

I viewed the retrospective plan for 13 November 2012. The actual plan cannot be provided through commercial in confidence. The retrospective plan was for November 2012 explaining the details of what is required to be discussed in retrospectives with an explanation of the usefulness of the planned topics and the approximate time required for each topic to be discussed.

Item 7: The team meeting minutes

I viewed the meeting minutes on 20 June 2012. The actual team meeting minutes cannot be provided through commercial in confidence however the meeting minutes was for 20 June 2012 and it contained the issues about the current tasks in hand and actions needed to be taken. It actually contained details that provide the foundation for the next meeting's agenda. For example, there was an issue that needed further discussion with the project owner and the team required to discuss about it in the next meeting.

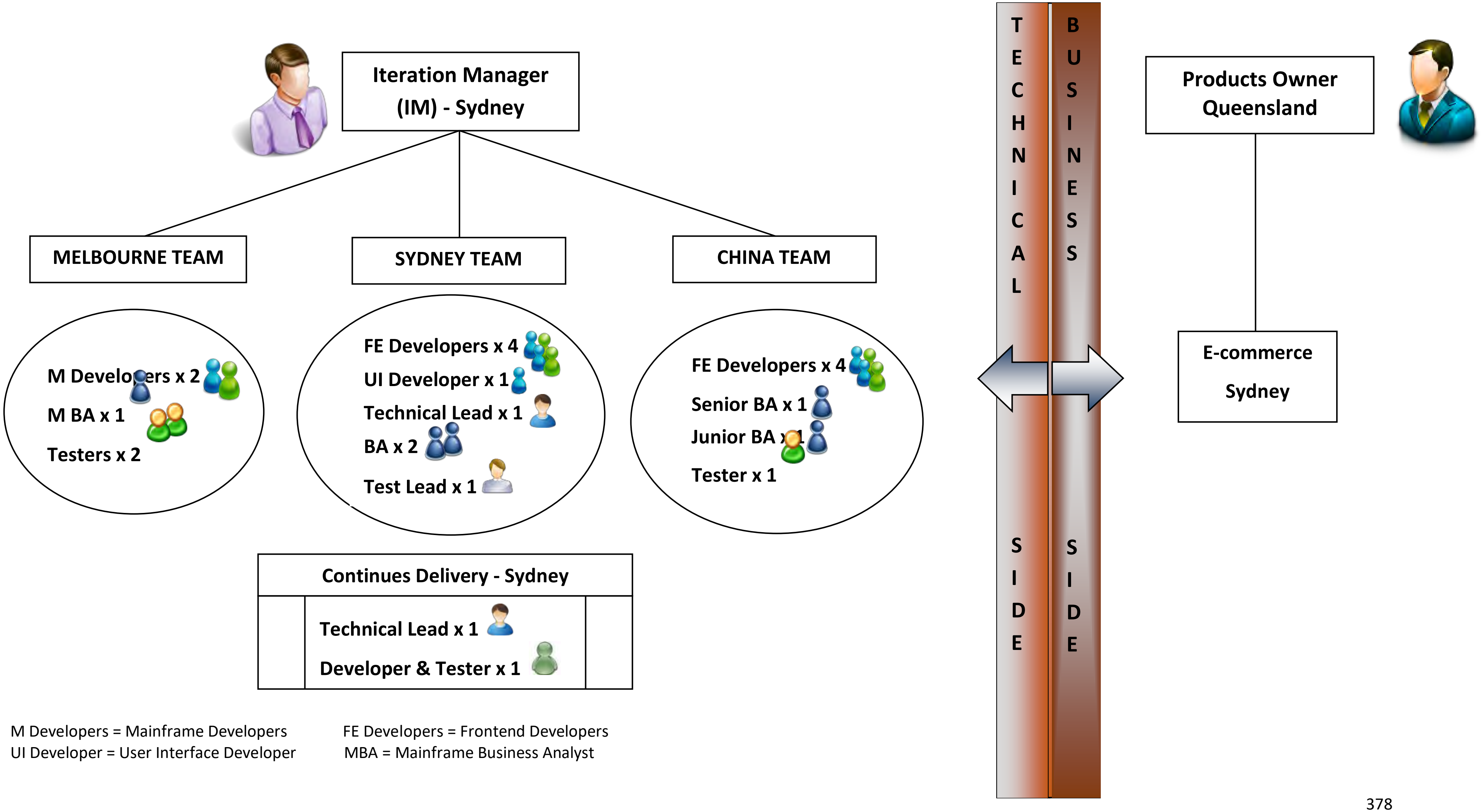
Item 8: Project performance report

I viewed the project performance report on 11 Sept 2012. The actual report cannot be provided through commercial in confidence however the performance report was for August 2012. The project performance report contained the analysis of past performance, current status of risks and issues, work to be performed in the next reporting period, significant achievements, forecasts and variance analysis. The report also provided details about the project changes that were identified.

Appendix J Overview of interview participants

Role	*Work and Method Experience*	*Sydney Team*	*China Team*
Iteration Manager (IM)	Employed with the company after graduation three years ago and started his experience with agile development. Previous role in the company was system engineer for one year and eight months; education background in computer science (software engineer Bachelor degree).	X	
Senior Developer	Employed with the company for more than a year as senior developer. Previously worked as analyst programmer in another company, education background bachelor of IT. Five years of experience with agile development methodology.	X	
Team Leader	Employed with the company for the last 11 years. Held various positions in the company, from senior developer, iteration management to project manager (5 years), and now team leader for the last year. Previous experience with agile as the infrastructure project manager for two years; educational background in economics and master degree in IT.	X	
Business Analyst	Employed with the company for more than a year from partner company as senior business analyst and at the same time coordinator and consultant to China team. Three years of experience in agile software development. Previous work experience as junior IT support for five years, as business analyst for about six years and also worked as iteration manager. The educational background is in network engineering, currently doing MBT degree.	X	X
Lead Business Analyst	Employed with the company for 10 years and two years of experience of agile software development. Working history of more than 20 years in IT industry as business analyst; educational background in business through TAFE.	X	
Project Manager	Employed with the company for 12 years, two years of experience with agile software development. Started out as a business analyst, experienced in a waterfall development environment for more than 15 years, educational background in business technology.	X	
Agile Coach	Employed with the company for more than 12 years and three years of experience with agile software development. In total worked as business analyst for 25 years in IT industry; education background in business through TAFE.	X	

Appendix K The Lake Project structure



Iteration Manager (IM) - Sydney

MELBOURNE TEAM

M Developers x 2

M BA x 1

Testers x 2

SYDNEY TEAM

FE Developers x 4

UI Developer x 1

Technical Lead x 1

BA x 2

Test Lead x 1

CHINA TEAM

FE Developers x 4

Senior BA x 1

Junior BA x 1

Tester x 1

Continues Delivery - Sydney

Technical Lead x 1

Developer & Tester x 1

TECHNICAL SIDE

BUSINESS SIDE

Products Owner Queensland

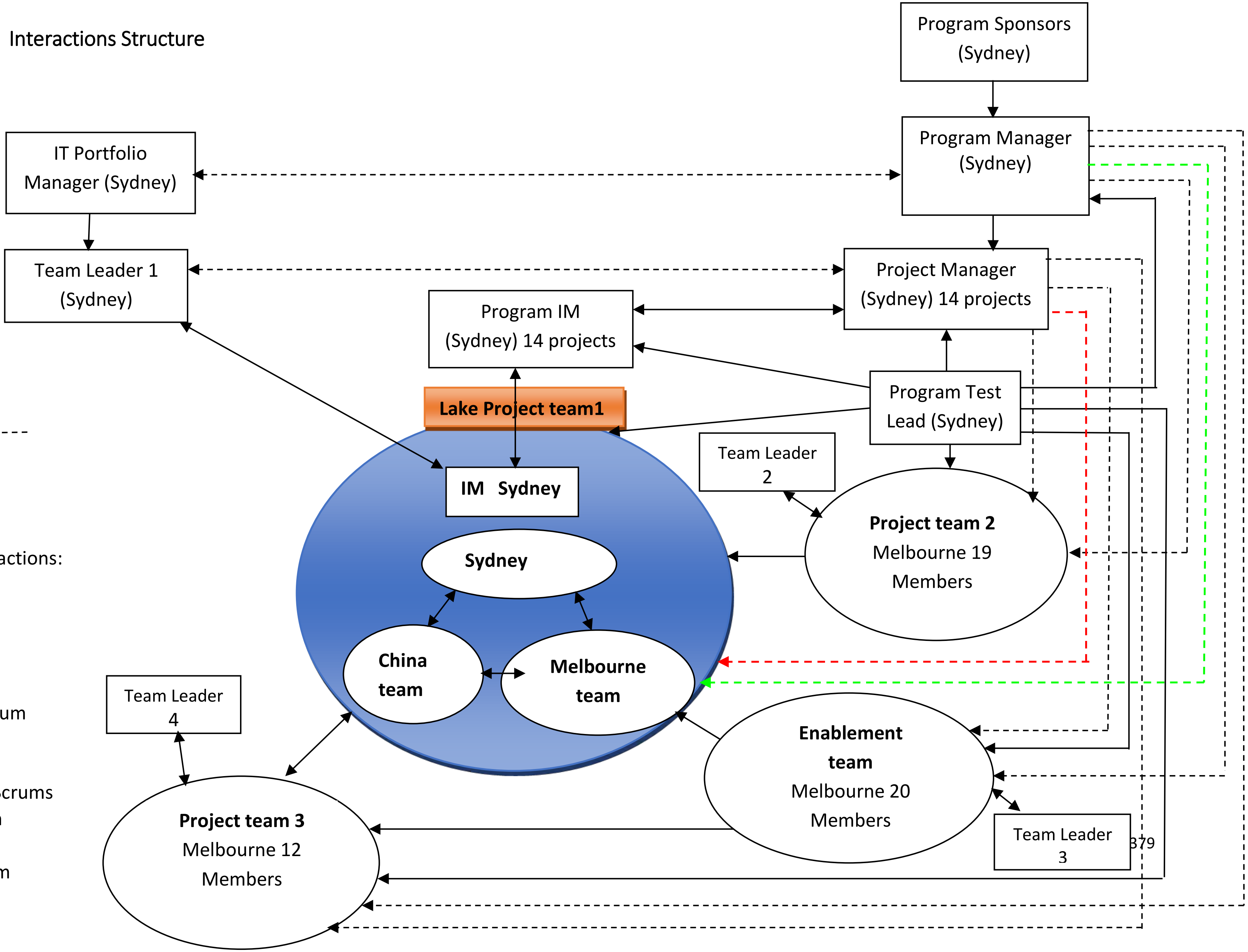
E-commerce

Sydney

M Developers = Mainframe Developers
UI Developer = User Interface Developer

FE Developers = Frontend Developers
MBA = Mainframe Business Analyst

Appendix L Interactions Structure



HR interaction ————

Informal interaction ------

The Lake Team interactions:

- IPM
- Showcases
- Stand ups
- Retrospectives
- Developer's Forum
- BA's Forum
- Tester's Forum
- Daily Scrum of Scrums
- Homemade Jam
- Tech Huddle
- Tech Lead Forum

Appendix M Roles Definitions

Roles	Definitions
Program Manager	Has the budget to resource the project to deliver. Within program A there is a Business Technology (BT) program manager, for all the change readiness and there is the IT program manager, for the IT part of the project. BT and IT program managers both report to program sponsor. BT program managers are accountable for the business technology delivery of all the streams. They are more about forward planning and an escalation point in a sense that they take care of escalated issues that the project managers cannot resolve because they have connections with business stakeholders to resolve bigger problems. IT program managers are responsible for the IT delivery of all the streams within the program. There is also a business transformation team on the business side that documents any new processes, the process system that they use for training, training material, coordinating the training, coordinating the business implementation. They also do a lot of business scenario testings. They interact and report to the program manager.
Program Sponsors	They are the key link between the project management and the organisation's executive management. They have got accountability on the business side as well as the IT side for delivery of the project. Program sponsors own the project and have the ultimate responsibility for seeing that the intended benefits are realised to create the value forecast in the business case. The program sponsors dedicate the budget to program managers and they need to make sure that the project is delivered according to the dedicated budget. They communicate on behalf of the project team, particularly with other stakeholder groups in senior management.
Program IM	The program IM is the coordinator between different IMs and program managers. The reason there is a need for program IM role is that there are a lot of projects involved in a single program that requires an additional point of contact to coordinate. The current program of work has 14 streams or 14 projects, and the program IM sits across all of those, manages issues and coordinates between project teams, project manager and program manager. The program IM reports some of the issues and problems of the project team to the project manager. This is a coordination role in the delivery sense and there is no direct hierarchical relationship in a HR sense between these two roles.
Program Test Lead	The program test lead is somebody who is responsible for making sure the way applications are tested across the projects (streams) within the program is consisted. This person communicates directly with all of the

	projects as well as heavy communication with the program IM, project manager and program manager. The program test lead reports to the project manager.
Project Manager	Responsible for all of the streams across a program in a business technology sense and accountable for delivery of a collection of related projects. This responsibility is sometimes shared with team leaders depending on the size and scale of the project. The smaller projects that take less amount of time will probably be run by team leaders but for larger and lengthy projects, project managers probably do part of the work as well. Project managers act as a coordinator between IMs and program managers and report to the program manager in a HR structure and to the related domains in the delivery structure. Team leaders' project status report goes to the related project manager. In addition, the program IMs report to project managers. Project managers have a strong view of the program and are responsible for actual delivery, making sure that individual project within a program is delivered.
Team Leaders	They are accountable for delivery and strategy of projects, making sure the projects within streams are running smoothly. They also ensure that the streams are delivered through their online strategy. Each team leader has people working in certain programs including the core team, they report to them on delivery structure but from a HR structure point of view they do not necessary report to the same team leader. Team leaders tend to be specialised in the business that portfolio managers are allocated to, but they may also look after delivery of another part of the business. Team leaders report to portfolio managers, they have accountability and responsibilities to their portfolio manager to ensure that the streams are successful from a delivery sense and strategy sense.
Business Analyst (BA)	Collect all the requirements, liaise with the business representatives and find out what needs to be done in terms of building features and functionality of the application.
Iteration Manager (IM)	A person who is embedded within the project core team undertaking the role to help plan and deliver the project and to collaborate with the rest of the project team. IMs make sure that all the core team members are running efficiently. They are responsible for day to day communication of the project team and run day to day activities such as set up meetings, manage story cards, make sure the wall is updated and following up task completion. They also make sure iteration's commitment will be fulfilled and point out bottlenecks in the delivery process. They are the central

	point that aids in the delivery of the project. IMs form groups of people (the core team) working together towards common goal meaning they succeed or fail together. They avoid noise and reacting to it if necessary. In a nutshell their role is basically to ensure that team is able to develop business value in the most efficient manner that they can. They report to the team leader.
Technical Leads	They are responsible for the quality of the application. Depending on the stream, technical leaders report to different people, for example team S technical leader reports to the delivery team leader. The technical leaders' role is to ensure that they have got an overview of the development work. In other words, they provide solutions to technical issues, and are responsible for meeting development schedules and ensuring the delivered solution meets the technical specifications and design requirements.
Test Leads	This role is basically to make sure that all the testing work is distributed amongst the testers in the project team and the test lead as well. In other words the test lead leads all the testers.
E-Commerce	They are business representatives responsible for interacting with the business about the raised issues and requirements relating to the business.

Appendix N Storyline Memos

Co-locating practice

It seems that the Lake project faced similar challenges like any other distributed project due to distance and being distributed across borders. One of these challenges is that after a while when team members are not involve in every aspects of the project they feel isolated and not being part of the team. One reason could be that due to cultural differences. So, the Lake project came up with the amazing practice called 'co-locating' the distributed teams that improved many challenges of the distribution aspects of the project.

The Lake Project co-located all the three teams from Sydney, Melbourne and China in Sydney for a short period of time. All the team members expressed the benefit of such practice from the first time they were able to communicate and meet face-to-face, the Lake project Coach says. The first co-locating practice was in Sydney at beginning of the project phase. It really helped the project team members from all the three teams to socialise and familiarise themselves with other team members. To help socialisation of the team members that have never met before, the Lake project organised socialising events and playing games sessions. This, later on, helped in their communication when they were back to their original sites. It was especially helpful for the Chinese team to get to know more about Australian working standards and culture.

During the co-locating practice, all the teams went through plenty of workshops. These workshops are more critical at the initial phase of the project because they help the teams to have a smooth start of the project. Indeed, the workshops put all the teams across geography on one page to have a smooth start, reduce variances in goals and expectations among the teams.

In other words, it improves the coordination among the teams. I understand that the project coach plays a key role during the workshops and the later follow ups with the team to resolve any issue that would be raised by the teams during the project life cycle.

One of the most important points that needs to be taken into consideration when practicing 'co-locating' teams, is to repeat this practice during the project life cycle. Indeed, to obtain full benefits of this practice, the teams need to be co-located during the project phases. After the first co-locating practice, Lake Project co-located the Sydney team in China and the Sydney learn as much and became familiar with Chinese working culture, logic and standards. It was very helpful when the teams communicating about the context and made it much easier for all the team members to makes sense of the context much easier.

Collective sanctions

If there is any unaccepted behaviour, they talk to the person. If the person is still not performing well, they escalate it to the HR leader. Also because they are on bounce system here it could have financial consequences for them, because you have to have a minimum successful level in the annual performance review. If you do not get to your successful level you do not get your bounce and no salary increases or promotions. And if you are overly performing you will get the bonuses, promotion and salary increase.

Beside the financial consequences, there are also non-financial consequences. For example, you might be spoken to, they try that no one to get negative feedback from the leader, but they try to resolve issues at the first level before escalating it. Agile is a team that they try to pull each other up especially with the high performance team, they do not like the not contributing members and do not have much tolerance for that. They try to create a culture where people are comfortable and they can pulled up by high performers ideally. So, if that does not work and if there is a bad behaviour and it does not get resolved then there will be a discussion at the team level and if it still remain the same then they will escalate the issue to the formal performance manager. But at the end of the day, the goal is to resolve the issues within the team level.

Trust and visibility

It seems there is a lot of trust among the teams otherwise without that the team would not be able to work that well. There is a lot of trust amongst the teams to deliver the task, to assign the task to them, to complete the task on time. Indeed, trust is one of the shared values among the teams and they have been living that value very much. When a task is assigned to a team member, they trust that the person complete the task on time and they won't be around and nagging when can they have it unless they need to coordinate something.

Although there is a lot of trust between teams but the visibility was lost at the beginning of the project phase. The team members trust that other member can complete the task but they always know that the other team member has a task until they do the check up throughout the day. In other words, because of the distance between the teams, the visibility was lost among the teams. The Lake project used a virtual tool called JIRA to put a safeguard around it to make sure that it does not happen again.

JIRA has a virtual view of all the tasks and the teams can get into it and find out which task is assign to which team member. However, what they did not do very well was they did not update the JIRA and as the result the task assignments were not updated and made confusion. They put a lot of effort to keep JIRA updated. The JIRA is updated as soon as a task is picked up by a team member. Anyone that pick up a task has responsibility to update the JIRA. The Lake Project had JIRA at the beginning of the project but they did not use it to its full functionality and properly. With keeping the JIRA constantly updated, there is a lot more transparency.

Appendix O Pilot Case Study Protocol

Potential Data Sources	**Construct**	**Questions of the Case**
All respondents	Individual demographic	1) How long have you been employed in this organization? 2) What is your title? 3) How long have you been in your present position in this organization? 4) How many years of experience do you have in your current role? 5) How long have you been a member of the project team you described in this interview?
Senior & executive managers	Business context *(Brief description of company)*	1) How many people work in IT department? 2) Where does the IT department fit into the organizational structure? 3) What is the strategy for IT department?
Project managers *Team members*	Team distribution *(Project background)*	1) What does the project team is working on? 2) Where are the project teams geographically located? 3) Have all members of the project team been part of the team since its inception? 4) What development method does project team use? 5) Do development teams use various methods on different sites? 6) What is the total number of people in your project team? 7) How do the project team members communicate across locations?

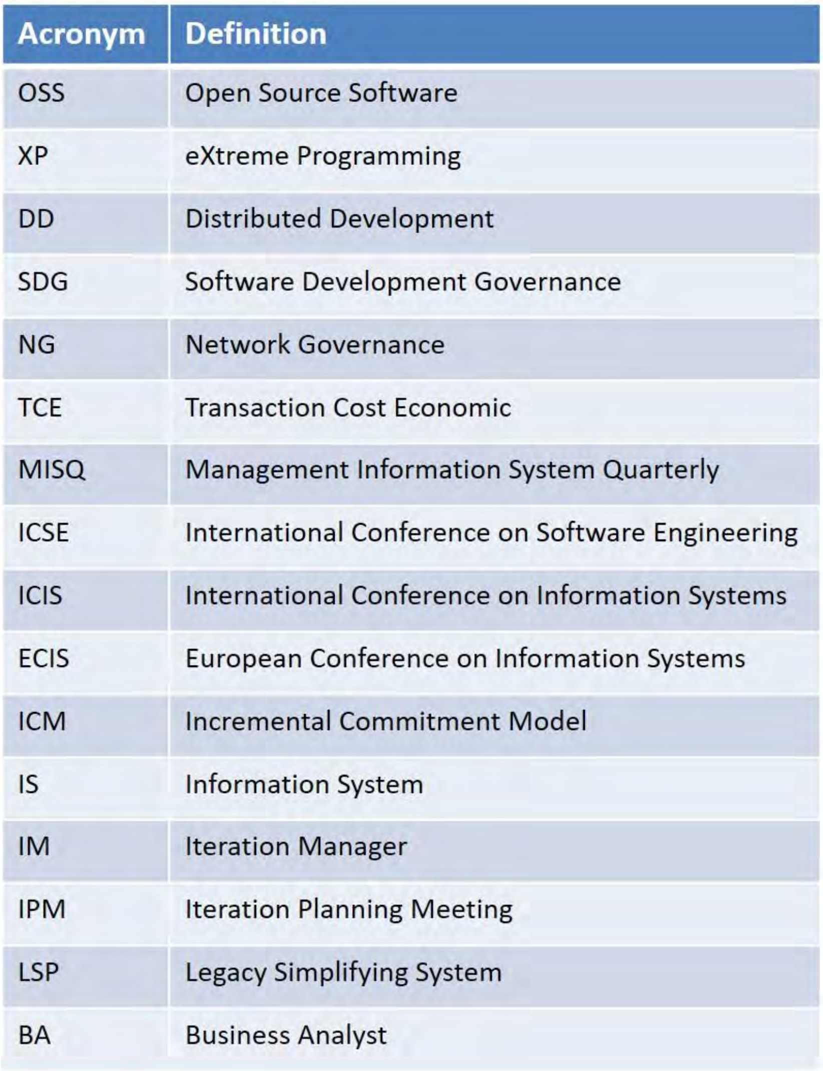
Potential Data Sources	**Construct**	**Questions of the Case**
Project Team members	Development method	1) Have the project team had specific training on agile methods?
	(General values & principals)	2) What is (are) specific agile method(s) used by the team? 3) Do the project team members recognize the agile methods as such? 4) What are the key agile's practices used across teams? 5) Do the project teams have worked on other agile projects?
Project managers	Macroculture	1) Do project team share a common understanding of the
Team members	*(Core Norms)*	agile norms and values across distributed teams? ▪ What is (are) the impact(s) of such share understanding on coordinating across the teams? ▪ Does such share understanding and believes have any impacts on protecting the interaction across distributed teams? ▪ Does such share understanding exit between business representatives across distributed teams? ▪ How about between developers across distributed teams? ▪ How about business representatives from one team with developers across other teams?

Potential Data Sources	**Construct**	**Questions of the Case**
		▪ Do distributed agile development project teams experience dissimilar ways of working at different sites? Please explain. ▪ In your opinion, did you find some agile values or principles too difficult to adhere to in distributed steams?
	Restricted Access (*Protect interruptions to work*)	1) Do you experience any type of interruption to your work resulted from distributed teams? Give me an example. ▪ Where are they coming from mainly? ▪ Who are involved? 2) Are there any restrictions across distributed teams? Give me an example. 3) Do such restrictions have impacts on coordination across distributed teams? How? 4 Do such restrictions have impacts on protecting interactions across distributed teams? How?
	Collective Sanctions *(Imposed punishments)*	1) What type of behaviours are not tolerated across distributed teams? Give me some examples. 2) How are these sorts of behaviours managed across distributed teams?

Potential Data Sources	**Construct**	**Questions of the Case**
	Collective Sanctions *(Imposed punishments)*	3) Do any of these disciplines apply to the business representatives across the teams too? How? 4) What are the impacts of applying the disciplines across the teams? 5) How do you get around the challenges? 6) Do imposing such disciplines have any impacts on coordination across distributed teams? How? 3) Do such disciplines have impacts on protecting interactions across distributed teams? How?
Project managers Team members	Reputation	1) How important is for team members to earn positive reputation across teams? 2) How team members earn such reputation across teams? ▪ Why is it important for them? 3) How do teams protect their reputation across teams? Please explain. 4) How teams become aware of the other teams' reputation across locations? 5) Does such reputation have protected interactions across teams? How?
Potential Data Sources	**Construct**	**Questions of the Case**

		6) Does team reputation have any impact on coordination across teams? How?
Project managers	Coordination *(Task coordination/ informal coordination)*	1) How do teams exchange information, knowledge or sharing skills across teams? 2) How often do distributed teams interact? 3) What do you encounter to encourage coordination across teams? 4) What do you do to get around coordination challenge?
	Safeguarding *(Protect interactions)*	1) What do project teams do to protect their interactions across teams? 2) Is there a conflict management strategy in place? Give me an example. ▪ How well does it work? 3) What are the main challenges you encountered across teams?

Appendix P Acronyms



Acronym	Definition
OSS	Open Source Software
XP	eXtreme Programming
DD	Distributed Development
SDG	Software Development Governance
NG	Network Governance
TCE	Transaction Cost Economic
MISQ	Management Information System Quarterly
ICSE	International Conference on Software Engineering
ICIS	International Conference on Information Systems
ECIS	European Conference on Information Systems
ICM	Incremental Commitment Model
IS	Information System
IM	Iteration Manager
IPM	Iteration Planning Meeting
LSP	Legacy Simplifying System
BA	Business Analyst