A Mixed Methods Investigation of Ethnic Diversity and Productivity in Software Development Teams

A thesis presented in partial fulfilment of the requirements for the degree of

Doctorate of Philosophy in Information Systems

at Massey University, Wellington,

New Zealand.

Julian Congalton

2014

ABSTRACT

Software has become pervasive across all aspects of society in the developed world and as a result, society has become highly dependent on new software being created for any modern advancement. Much research has focused on reducing the cost to develop software, including understanding what makes software teams more productive. Software teams are increasingly ethnically diverse due to the growth in distributed software development and a globally mobile labour force. Team composition has been found to be a major influence of team performance and ethnic diversity in teams can improve innovation and problem solving. As software development relies on effective teams and often involves solving complex problems, this raises the question of how ethnic diversity within software development teams affects the performance, and therefore productivity of those teams. This research seeks to understand how ethnic diversity in software development teams influences the productivity that those teams achieve. This is important as software related costs represent a significant component of business costs. Furthermore, the cost effective development of new or changed software is critical to support advances in today's technology-dependent society.

A mixed methods research approach has been used in this study with an emphasis on qualitative data. This is the first mixed methods study of productivity in New Zealand software development projects and represents a unique examination of the sociological effects of ethnic diversity in software projects. Using a conceptual model of software development as a socio-technical system, project documents and interviews with project managers were analysed. A detailed analysis reveals themes and patterns regarding the influence of ethnic diversity in software development productivity. The qualitative data has been complemented with quantitative analysis of the project data using the productivity model embodied in the software development cost estimation model COCOMO II combined with indices measuring ethnic diversity. Ethnic diversity improved team problem solving and

innovation on complex software projects but hindered some aspects of communications which negatively influenced productivity, particularly on large projects. Ethnic diversity could either enhance or impair team cohesion, depending on whether the project manager took steps to build relationships and trust within their team.

ACKNOWLEDGEMENTS

Many people have contributed to this thesis including those who made it possible, those who encouraged me to start it, and those who have supported me during the process. I would first like to thank my supervisors, Keri Logan and Barbara Crump whose patience and guidance enabled me to complete this journey. Their tireless support and encouragement has made this thesis possible.

I wish to thank the project managers and organisational leaders who agreed to participate in this study and made it an enjoyable and insightful process. It is only through their generosity and trust that this thesis has been feasible.

My thanks goes out to all the managers and mentors I have worked with, and for, during my 20 years in the software development industry. Many of these people have encouraged me to continue studying while working. In particular I would like to thank Alan Murphy who, as my manager for the five years before I began this thesis, supported me in completing the study that led up to, and enabled me to undertake this PhD.

I am grateful to my father, my mother and my sister Jenna who were always interested in what I was researching and keen to read my thesis. I am also thankful to my grandfather, Professor Athol Congalton, who, at an early age encouraged me to ask questions about belief, truth and knowledge.

Finally, I am eternally indebted to my wife Donna who never wavered in her support of me undertaking this thesis. Without her support I could never have begun this journey and would never have completed it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 – INTRODUCTION

Software has become pervasive across all aspects of society in the developed world (Asmild, Paradi, & Kulkarni, 2006; Charette, 2005). As a result, society has become highly dependent on new software being created for everyday devices such as ATMs, cell phones and cars (Boehm, 2005; Charette, 2005; Tan et al., 2009) as well as life supporting systems such as healthcare (Panousopoulou, Galway, Nugent, & Parente, 2011; Schrenker, 2006; Skevoulis, Campedelli, Holdsworth, Verel, & Tavales, 2009), water supply (B. L. Smith, 2002; The Economist, 2011) and the provision of electricity (Kangilaski, 2009; Martinez, 2008; Sinha, Lahiri, Chowdhury, Chowdhury, & Song, 2007). New business initiatives are dependent on new software or changes to existing software (Banwet, Yadav, & Momaya, 2003; Boehm, 2006; Shee & Pathak, 2006). The ubiquity and dependence on software applies in the government sector where new legislation can often only be enacted as quickly as the affected software can be updated, tested and implemented. For example, in the major 2010-2011 reorganisation of Auckland’s local government into a super city, the majority of the costs related to the changes required to information technology systems (Orsman, 2010). However, improvements in software development productivity have not kept pace with technological advances in hardware (Anthes, 2005), society’s increased dependence on software (Mens et al., 2005) or the general growth in demand for more software (Pressman, 2010). The effect of this sub-optimal productivity is compounded by the fact that the skills required for software development command a high salary or hourly rate (Boehm, Penedo, Stuckle, Williams, & Pyster, 2007; New Zealand Career Services, 2010). The high cost of software creates barriers to implementing advances in today’s technology dependent society.

Much research and many industry initiatives have focused on reducing the cost to develop software and improving productivity. These include process improvement methodologies (Bonacin, Baranauskas, & Rodrigues, 2009; Carnegie Mellon Software Engineering Institute, 2006; Gibson, Goldenson, & Kost, 2006; McGuire, 1996; Montangero, 1999; Sommerville, 2004), moves to agile development approaches (Beck, 2000; Cockburn, 2007; Fowler, 2005; Highsmith, 2002; Martin, 2003) and outsourcing to offshore software development organizations (Boden, Nett, & Wulf, 2009; Ehrlich & Chang, 2006; Herbsleb & Mockus, 2003; Herbsleb & Moitra, 2001; Šmite, Wohlin, Gorschek, & Feldt, 2010). However, researchers consistently report that regardless of the software development method utilized, a well formed and effective team is critical for productive software development (Alleman et al., 2004; Boehm & Turner, 2005). An effective software team is one which has attributes such as creativity (Chatenier, Verstegen, Biemans, Mulder, & Omta, 2009; Post, De Lia, DiTomaso, Tirpak, & Borwankar, 2009; Tadmor, Satterstrom, Jang, & Polzer, 2012), collaboration (McCreery & Moranta, 2009; Stober & Hansmann, 2009) and good communication (De Farias Jr, De Azevedo, De Moura, & Da Silva, 2012; Pikkarainen, Haikara, Salo, Abrahamsson, & Still, 2008; Santos & Moura, 2009). These attributes can be found, if well managed, in ethnically diverse teams (P. Richardson, 2005; Shachaf, 2008; Winkler & Bouncken, 2009). For this reason it is important to examine the influence of ethnic diversity in software development teams. However, there is little or no existing research into the influence of ethnic diversity on the effectiveness of software development teams, which also takes into account other factors already known to affect software development productivity.

Within the area of software development and in other disciplines, it has been shown that team composition is a critical determinant of team performance (Faraj & Sproull, 2000; Guzzo & Dickson, 1996; Katzenbach & Smith, 1993; Woolley, Gerbasi, Chabris, Kosslyn, & Hackman, 2008). Software development teams have traditionally been relatively homogeneous, being composed primarily of people who are white and of western culture (Trauth, Quesenberry, & Huang, 2006).

Now, immigration, greater labour mobility and globally distributed projects mean that software teams are more likely to be ethnically diverse (Badkar & Tuya, 2010; International Labour Organisation, 2008; Šmite, et al., 2010). In the 2013 New Zealand census it was shown that one in four New Zealanders are now born overseas and the Asian population in New Zealand has doubled since 2001 (Statistics New Zealand, 2013) and this diversity is represented in the workforce. There is a shortage of people with the skills required for software development in New Zealand (New Zealand Ministry of Business Innovation & Employment, 2013). New migrants often gain entry to New Zealand and find jobs because of their technical software development skills. Software teams are also increasingly ethnically diverse due to more collaborative, free and open source development which typically occurs at many global locations (Chatenier, et al., 2009; Shibuya & Tamai, 2009) and the global outsourcing of software development (Šmite, et al., 2010; Winkler & Bouncken, 2009). Furthermore, ethnic diversity is becoming common in software teams as they are often geographically distributed, meaning members of the team are working in different countries (Mathieu, 2009). This changing composition of software development teams suggests that it is important to examine the influence of ethnic diversity in these teams.

The reasons why ethnic diversity can affect team performance are best explained by examining the concepts underpinning ethnicity. Early literature supported beliefs that biology determined many characteristics associated with race, but more recent literature has criticised this “biological essentialism” (DeCecco & Elia, 1993, p. 2). Instead, ethnicity is now described as being, at least in part, a social construction (Cormack, 2010; Norval, 2004). While ethnicity has its origins in biological race, socially constructed culture is also an important component as this includes values, behaviours and how individuals interact (E. T. Hall, 1983; Winkler & Bouncken, 2009). Some information systems literature has focused on national culture to capture differences in values and beliefs (Ali & Brooks, 2008; Earley & Mosakowski, 2000; Hofstede, 1980a) but it has been argued that ethnicity is a better indicator of values and beliefs as this transcends national boundaries (Eriksen, 2003; Myers & Tan, 2002; A. Smith, 1986). Ethnicity reflects critical aspects of an individual's values, beliefs and therefore behaviours, and has been found to influence team performance (Brandes, Franck, & Theiler, 2009; P. Richardson, 2005; Watson, Kumar, & Michaelsen, 1993; Winkler & Bouncken, 2009).

Research into ethnicity in New Zealand has focussed on the identity of Maori, the indigenous people (Kukutai, 2007; Sibley, Liu, & Khan, 2008) and reported on ethnicity issues in New Zealand for significant immigrant populations such as people from the Pacific region (R. S. Hill, 2010) and Asia (Badkar & Tuya, 2010). More recently, it has been found that many qualified immigrants to New Zealand of different ethnicities are unable to secure employment in the jobs for which they are qualified (Ward & Liu, 2012). New Zealand has a "mature software development industry" (Investment New Zealand, 2007, p. 3) with software development worth NZD 3 billion in 2008 (Statistics New Zealand, 2009) and a 78 percent increase in software jobs from 2000 - 2010 (Statistics New Zealand, 2012). The combination of the increased ethnic diversity in New Zealand and the growing software industry gives rise to the need to improve understanding of how ethnic diversity affects software team performance in New Zealand.

A number of studies have been undertaken to analyse the performance of software development teams (Humphrey, 2000; McGuire, 1996; Moe, Dingsøyr, & Dybå, 2010) and a few have considered the effect of diversity on such teams. Erdogmus (2009) highlighted that team diversity can improve problem solving in software development teams. Liang, Liu, Lin and Lin (2007) examined how knowledge and value diversity (both associated with ethnic diversity) affected software teams, finding that knowledge diversity helped team performance but value diversity hindered it. Some studies have presented frameworks for analysing diversity (Dafoulas & Macaulay, 2001; Walsham, 2002) but do not report on any actual research undertaken. In those studies that report on empirical research on software development teams, diversity is analysed along dimensions such as personality diversity (Pieterse, Kourie, & Sonnekus, 2006), values diversity (Liang, et al., 2007), information diversity (Liang, Jiang, Klein, & Liu, 2009) and cultural diversity (Borchers, 2003; Egan, Tremaine, Fjermestad, Milewski, & O'Sullivan, 2006; Niazi, Babar, & Verner, 2010; Walsham, 2002). While the dimensions used in some cases relate to ethnic differences, the influence of ethnicity on the performance of software teams has not been widely analysed. Furthermore, existing studies do not consider other major factors such as product complexity, requirement changes and programmer capability, which are known to affect software development productivity. While making useful contributions to the body of knowledge, previous studies on diversity in software development teams do not report on the influence of ethnic diversity on software development productivity. This leaves a gap to be filled through mixed methods research into the influence of ethnic diversity on software development productivity, taking into consideration other factors that are known to affect productivity.

Research Questions

This study seeks to understand the influence of ethnic diversity on the productivity of software teams. A review of the relevant literature has identified that a key factor affecting productivity is team composition and in particular ethnic diversity. However previous studies generally report that there is no direct relationship between ethnic diversity and team performance and instead ethnic diversity can influence performance through various team processes and other intervening variables. Therefore, in order to understand the influence of ethnic diversity in software teams, it is first necessary to understand the factors that influence productivity and then examine how ethnic diversity influences those productivity factors. This gives rise to the first two research questions:

1. What factors influence productivity in ethnically diverse software teams?
2. Does ethnic diversity influence the productivity of software teams (through the factors that influence productivity)?

The influence of ethnic diversity on team performance has been found to be affected by mediating factors such as organisational context which leads to the third research question:

3. What mediating factors alter the influence of ethnic diversity on the productivity of software teams?

Overview of the Research Design

A mixed methods design using qualitative and quantitative approaches has been used to gather and analyse data about historical software development projects at software producing organisations. Mixed methods have been found to be useful in a number of areas of research (Petter & Gallivan, 2004) including in the field of information systems (IS) research (Mingers, 2001). For example, Ormerod (1995) used mixed methods to gain multiple perspectives on the process of IS strategy development. More recently Williams (2009) used mixed methods to investigate the relationship between internet access and social cohesion, noting that this enabled data to be cross checked and provided multiple perspectives which improved the authenticity of the research.

While both qualitative and quantitative methods are being used in this study to provide richer and broader results, the research paradigm adopted is primarily qualitative. Caracelli and Greene (1997) argue that the use of mixed methods is a legitimate and effective means of generating “more relevant, useful and discerning inferences” (p. 19) from research. The research questions for this study indicate a qualitative paradigm, focusing on the influence rather than the effect of ethnic diversity. There are, however components to the research questions which imply a quantitative paradigm, in particular with examination of productivity. The Constructive Cost Model (COCOMO II) has been used in this study as this provides a comprehensive model of software development productivity based on quantitatively assessing the factors found to affect productivity.

Some mixed methods researchers suggest identifying a dominant method in a mixed method design (Creswell, 2009; Greene, Caracelli, & Graham, 1989; Leech & Onwuegbuzie, 2009) however others prefer to focus on a “theoretical driver” as this “implies the guiding of the research projects rather than one method being ‘better’ than the other” (Morse & Niehaus, 2009, p. 11). It is therefore appropriate to consider this study as primarily interpretive and adopt a qualitative paradigm as the theoretical driver, using interviews and document analysis. Combining paradigms and methods in this way has been used effectively in IS research to synthesise positivist and interpretive research conducted in parallel (Trauth & Jessup, 2000).

The use of methodological triangulation contributed to the credibility and validity of the research as this helps to provide a more in-depth understanding in interpretive studies (Denzin, 1997). Triangulation is the use of multiple methods or perspectives to analyse phenomena and can help to increase the validity of findings (Greene, et al., 1989). This technique has its origins in navigation where multiple readings are used to determine an exact location (H. W. Smith, 1975) and has been used in social sciences to improve the completeness and validity of research results (Bryman, 2004). While this study has some positivist aspects to it, its theoretical basis is primarily interpretive. Methodological triangulation through the use of qualitative and quantitative methods which has allowed the weaknesses of each method to be countered by the strength of the other (Denzin, 1997; Jick, 1979; Mathison, 1988).

Twenty four software producing organisations from across the North Island of New Zealand were approached and invited to participate. Of the 24 organisations approached, one declined, three initially accepted but later withdrew and 13 did not reply. Seven software producing organisations from Wellington, New Zealand agreed to participate – three government organisations and four non-government organisations. Of the four non-government organisations, two were primarily software producers, that is, their main activity was producing software. From these participating organisations, data were gathered on 19 software projects completed in 2011 and 2012 using interviews with project managers combined with a review of existing project and system documentation. Each project studied had a different project team. Of the 19 project teams studied, 14 were ethnically diverse, and one, which was also the smallest team, was not ethnically diverse. In the remaining four projects, it was not possible to gather data on the ethnicity of the project team members. The data relating to the 14 teams that were ethnically diverse have been used for the qualitative analysis while the data relating to all 19 projects were included in the quantitative analysis of the key factors influencing productivity.

Primary data was obtained using semi-structured interviews with software development project managers. Project managers were interviewed as they have extensive interaction with their team, play a key role in determining the project approach (Sebt, Shahhosseini, & Rezaei, 2010) and have a good understanding of the factors that affected the productivity of a project (Ehsan et al., 2010; Wang, 2009). The interviews were supplemented by a review of system and project documentation. The system documentation provided information about the system that was developed or enhanced, such as purpose, size and complexity. Project documentation provided further information such as total project effort, project duration and the challenges that were encountered during the project. Approximately 650 project and system documents were received and analysed to assess whether they could contribute to answering the research questions.

Due to the density of data generated in interviews, sample sizes for interview based research tend to be relatively small (Todd & Benbasat, 1987). In predominantly qualitative studies such as this one, there is a trade off between saturation of information, “where you’ve heard the range of ideas and aren’t getting any new information” (Krueger & Casey, 2009, p. 21) and achieving a representative sample (Teddlie & Yu, 2007). G. H. Hofstede, Neuijen, Ohayv and Sanders (1990) also comment on striking an appropriate balance in determining sample size to provide sufficient data for both qualitative and quantitative analysis. They comment that twenty teams from ten different organisations was a small enough number to allow an in depth qualitative study but also enable statistical analysis across the data from all teams (Hofstede, et al., 1990).

Significance

This study is significant for the following reasons. As far as can be ascertained, this is the first study of software development productivity involving ethnically diverse teams, in New Zealand. Second, the results of this research are useful for project managers and human resource managers when determining the composition of software development teams. Where software development teams are ethnically diverse, it is important for project managers to use the appropriate information and advice for maximising the benefits of ethnic diversity. Given that managers currently have little or no understanding of how to deal with diversity (Korn / Ferry Institute, 2013), the results inform software project managers of the benefits and risks associated with ethnically diverse teams.

Third, the mixed method approach addresses the “dearth of mixed methods research in information systems” (Venkatesh, Brown, & Bala, 2013, p. 1) . The inclusion of both qualitative and quantitative data enabled other factors already known to affect software development productivity such as requirements volatility, programmer experience and team continuity to be measured, and to assess how such factors were influenced by an ethnically diverse team.

Finally, this study contributes to mixed methods literature by producing meta-inferences and thereby going some way to reduce the "contribution shrinkage" (Venkatesh, et al., 2013, p. 11) that occurs when papers from mixed method studies are published based on individual methods with no synthesis. Examination of diversity in software teams has previously been studied either using only quantitative (for example, Egan, et al., 2006; Liang, et al., 2007; Pieterse, et al., 2006) or qualitative methods (for example, Borchers, 2003; Shachaf, 2008; Walsham, 2002). This study answers the call for greater use of mixed methods in IS research (Mingers, 2001; Venkatesh, et al., 2013) as the approach provides a more holistic understanding of complex systems such as the software development process.

Overview of Thesis

This thesis is organised into six chapters including this one. Chapter 2 presents a review of the literature most relevant to the study. This begins with a discussion of society's increasing dependence on computer software for everyday life as well as for advances in the standard of living. A discussion of the literature shows that this increased dependence on software has led to demands for more efficient software production and the approaches that have been taken to improving software development productivity. The review of the literature shows that team composition, and in particular ethnic diversity is an important factor affecting team performance. This is followed by a critique of previous research into diversity in software teams. The literature review explores the concepts of ethnicity and how ethnic diversity is an important aspect which can influence the performance of a team. The final section reviews how software development productivity is measured and improved.

Chapter 3 explains the method used to conduct this study. The theoretical framework is outlined, including the relevant theories, the conceptual model and the justification for the methods selected. An explanation is provided of how validity, reliability and transferability have been addressed in this study. The sample selected for this study is described, including a summary of each of the software development projects investigated. This is followed by a description of the instruments selected for this study and the data collection process including an overview of the ethical considerations addressed, along with the pretesting that was performed. Finally an explanation is provided of the mixed methods approach used to bring together the results and provide answers to the three research questions.

Chapter 4 reports the results obtained using the qualitative data to identify major factors affecting productivity in software projects and the influence of ethnic diversity. Thematic investigation of the interviews is used to examine the project managers' view of the major influencers of software development productivity. Once the most significant influencers of productivity are identified, the second part focuses on the influence of ethnic diversity on software development productivity.

Chapter 5 presents the quantitative analysis of the major influencers of productivity in the software projects studied and focuses on the correlations between the productivity factors and the diversity-related variables. Finally, Chapter 6 describes the synthesis of the qualitative and quantitative results, to elucidate and present meta-inferences arising from the mixed methods approach. The key findings relating to the research questions are summarised and the strengths and limitations of the study considered. The chapter concludes the thesis by commenting on the implications for theory and practice and finishes with recommendations for further research.

CHAPTER 2 – LITERATURE REVIEW

Introduction

This chapter presents a review of the literature relevant to this study, beginning with a discussion of society's increasing dependence on computer software for everyday life, followed by an examination of how that increased reliance on software has led to demands for more efficient software production. The chapter then explores factors that have been found to affect productivity, including the personnel and team related factors such as team cohesion, composition and diversity. This leads to an examination of the influence of ethnic diversity on team performance beginning with a review of literature on the drivers for workplace diversity and the effects of diversity in teams. A review of how organisational context affects the outcomes arising from diversity and some key differences between public and private organisations are discussed. Previous research into diversity in software development teams is reviewed.

The chapter then moves to focus on ethnicity and ethnic diversity as this is a critical aspect of diversity which has been shown to affect team performance and outcomes. The definitions and dimensions of ethnicity are presented, along with the different cultural values associated with ethnic diversity. As this study is conducted within New Zealand, issues and research on ethnicity in a New Zealand context are examined and reviewed. Previous research into the effects of ethnic diversity is discussed including the influence on innovation, conflict and communication. The final section of the chapter examines how productivity is measured, and existing models of software development productivity are discussed. This includes an introduction to the Constructive Cost Model (COCOMO II) which is used in this study as the basis for the conceptual model.

Literature Search and Review Strategy

This section outlines the search and review strategy by describing the search terms used, the article databases searched, inclusion/exclusion criteria, publication types, date range, and the review process. To locate previous research on the topics relevant to this study, a series of leading article databases were searched using a collection of search terms. These searches were repeated at regular intervals from 2010 through to 2013 and the results included both journal articles and conference proceedings. Alerts were set up on key databases using a subset of the core search terms so new relevant articles could be considered as they were published. The search terms and article databases used are listed in Table 1.

Table 1. Search terms and article databases

Subject	Terms used	Article Databases
The importance of software	• Software economic • Software expectations • software demands • software industry	Business Source Complete, Springer, ACM, Google Scholar
Software development	• Software art creative • Programming as an art • Creative software development engineering • Software product engineering • Software engineering product • Agile software development • Software intellectual • Intellectual property • Copyleft	Business Source Complete, Scopus, IEEE, Google Scholar, ACM
Software development productivity	• Programmer productivity • Software team productivity • COTS estimating • COCOMO • COCOMO tailoring • Software tools • Economies of scale • Software development productivity team • Software development productivity communication	IEEE, ACM, Google Scholar, Springer

Subject	Terms used	Article Databases
Productivity	• Productivity • Improving productivity • Labor productivity • Labour productivity	Palgrave, Business Source Complete, Web of Science, EconLit, Google Scholar, AEA
Project Management	• Project manager • Project management	EBSCO, Google Scholar, ACM, IEEE
Software teams	• Software development team • Software team • Agile software development team	Google scholar, JSTOR, ACM, IEEE, Business Source Complete, Springer and Scopus.
Government / non-government productivity	• Government productivity • Compar* public private productivity • Software productivity government • Comparative studies ISD practice • Differences in productivity between public and private software development projects • Public and private sector differences	Google Scholar, ACM, EBSCO, Scopus, IEEE, JSTOR
Diversity	• Diversity and team performance • Workplace diversity • Team diversity • Organisational context of diversity • Diversity in software development • Innovation • Conflict • Communication	IEEE, ACM, Google Scholar, Springer, JSTOR
Diversity in software teams	• Team diversity software development	Scopus, EBCSO, IEEE, ACM
Ethnicity	• Culture in IS • Cultural values • Conceptualizing culture • National culture • Ethnic • Ethnic diversity • Definitions of ethnicity • Ethnicity in New Zealand	Web of Science, Business Source Complete, JSTOR, Google Scholar
Diversity in New Zealand software teams	• (Diversity OR ethnic*) software development productivity • Software development (productivity OR efficien*) Zealand	Google Scholar, IEEE, ACM
Measures of ethnic diversity	• (((Ethnic OR race OR culture) AND (diversity)) AND (measur* OR index)) AND (organisation OR management OR business)	Google Scholar, EBSCO, Scopus, JSTOR

Many thousands of articles were returned by these searches. Approximately 2000 articles appeared from the initial assessment to be potentially relevant and these were imported into an Endnote database. From this shortlist, approximately 500 were used to support the literature review. In most cases, only articles published since the year 2000 were included, in order to ensure the information was relatively current. An exception to this was when articles appeared to be foundation articles for a particular area of knowledge. In addition to this systematic search process, there was also an element of organic discovery whereby one useful article led to other related material. For example, when an article was found which was highly relevant to this study, the publications cited in that article, along with articles which cited the original publication were also reviewed to search for other relevant information. Journal articles and conference papers were included in the literature search as well as grey literature where appropriate. Grey literature was used where publications such as industry and government reports were the most appropriate source of data. In some cases grey literature was referenced in the academic articles reviewed and in other cases it was found through the use of non-academic search engines (such as Google). The remainder of this chapter presents and critically analyses the literature found through the search and review process.

The Demands for Improved Software Development Productivity

Rising Expectations of Software

Society has become dependent on software as it is a critical component of the technology that underpins everyday life. Without software, devices such as ATMs, cell phones and cars would not be possible. In 2005 a typical cell phone had two million lines of software code (Charette, 2005) and by 2010 that had grown to over 12 million lines of code in the base operating system of many phones before any additional applications are added (Canalys, 2011; Leon, 2010). While a premium car like a Mercedes may have over 100 million lines of software (Charette, 2009), even a basic car has around 10 million (Broy, 2006). Healthcare is another example where society depends on software to help humans live longer and healthier lives (Skevoulis, et al., 2009). A number of aspects of healthcare cannot operate without effective software, including diagnosis of illness, management of medical conditions and administration of patient records (Charette, 2005; Panousopoulou, et al., 2011; Schrenker, 2006). As well as individuals depending on software, it is also important to the operations of most organisations. Software is a critical aspect of most new technology and technology in general is a key factor in determining most organisations' competitiveness. It has been argued that technology innovation is the most important factor in providing competitive advantage for business (Banwet, et al., 2003; Shee & Pathak, 2006). As early as 1989, Ramo (1989) predicted that the twenty-first century would be dominated by technology and technological competitiveness. Software is at the heart of modern technology and Boehm (2005) contends that:

> Software is increasingly becoming the most critical success factor for future products (automobiles, aircraft, radios) and services (financial, communications, defence). It provides both competitive differentiation and rapid adaptability to competitive change. It facilitates rapid tailoring of products and services to different market sectors, and rapid and flexible supply chain management. (p. 12)

Over the last 50 years the demand for more complex software has increased although the time available to create new software has reduced (Santos & Moura, 2009). This has led to some advances in the way software is developed but organisations still face problems with software development productivity. Software projects are growing larger in size and becoming continually more complicated in order to fulfil society's needs while at the same time delivery demands require faster turnaround time (Boehm, 2006). It is not just the complexity of software that is increasing, but also the quantity required and this growth is expected to continue for the foreseeable future (Tan, et al., 2009). In the context of this constantly increasing demand, software developers are generally capable of producing good software, but the real problem is how to produce software fast enough to meet the growing demand for it (Pressman, 2010).

Another indication of the growth in demand for software is the economic value of the software industry and its rate of growth. The value of the software industry worldwide in 2010 was estimated by market researcher Research and Markets to be USD 265.4 billion (December 2011) and by Gartner to be USD 244 billion (Gordon et al., April 2012). According to Research and Markets (2011) the value of the worldwide software industry in 2010 increased by 7.1% compared to 2009 and they forecast that in 2015, the global software market will have a value of USD 356.7 billion, an increase of 34.4% since 2010. Statistics analysed by Sawyer (2001) on historical data from 1986 to 1995 show similar growth trends in the software development market. The size of the whole IT market is significant and software development is a major component. Charette (2005) elaborates on the significance of investment in software by explaining:

> IT is now one of the largest corporate expenses outside employee costs. Much of that money goes into hardware and software upgrades, software license fees, and so forth, but a big chunk is for new software projects meant to create a better future for the organization and its customers. (2005, p. 45)

Both in New Zealand and on a global scale, software production is a significant activity which is growing each year (Gordon, et al., April 2012; Research and Markets, December 2011; Statistics New Zealand, 2012). In New Zealand, the value of software sales by organisations which have recorded their primary activity as "computer system design and related services" was approximately NZD 3 billion in 2008 (Statistics New Zealand, 2009). This does not include software development by organisations for which computer system design is not their primary activity (Statistics New Zealand, 2009) such as banks, insurance companies, manufacturers and government agencies, many of which have their own software development teams. Therefore, the actual expenditure on software in New Zealand is likely to be considerably higher. As a result, improvements to the efficiency of software development benefit a great many organisations.

Drivers for Efficiency in Software Production

Efficiency is generally defined as the ability of someone or something to achieve a specific purpose (Oxford English Dictionary, 2013). In economic studies, efficient production is described as the point at which all inputs to, and outputs from a production process can no longer be improved or optimised (Charnes, Cooper, & Rhodes, 1978; Cooper, Seiford, & Zhu, 2011). While there are a number of ways to measure production efficiency, productivity is widely used and is expressed as a ratio of inputs to outputs (Cooper, et al., 2011; Farrell, 1957). In this study, 'efficiency' is used as a general term, to refer to the ability of a team to achieve a specific purpose whereas 'productivity' is the more specific ratio of inputs to outputs. The definition of software development productivity is underpinned by the general definition of productivity which is:

> The amount of output per unit of input achieved by a firm, industry, or country. This may be per unit of a particular factor of production, for example labour employed, or per unit of land in agriculture, or 'total factor productivity' may be measured, which involves aggregating the different factors. (Black, Hashimzade, & Myles, 2009)

Given that most software development is undertaken by teams as a project (Franke, Narman, Hook, & Lillieskold, 2010) one way to describe software production efficiency is to consider definitions of success for software projects. A project is defined as "an endeavour with defined start and finish dates undertaken to create a product or service in accordance with specified resources and requirements" (ISO/IEC, 2008, p. 5). Standard definitions of project success focus on the "triple constraint" of "time, cost and specifications" (Meredith & Mantel, 2011, p. 3). That is, projects which are delivered on time, within budget and which meet the specified requirements are considered successful. However, the understanding of project success has changed and evolved over time (Jugdev & Muller, 2005) and simply meeting the time, cost and quality measures is generally no longer sufficient. Broader definitions of success in software projects encompass the triple constraint but include other factors such as stakeholder satisfaction, project team satisfaction and quality (Thomsett, 2002).

Success for software projects has also been shown to be a subjective concept and differs between countries. For example, “significant differences in definitions of project success” (Pereira, Cerpa, Verner, Rivas, & Procaccino, 2008, p. 905) were identified between US and Chilean software developers. US teams were more likely identify project success where they produced an easy to use product that the met requirements while Chilean team members were more likely to equate project success to accurate estimates and keeping to schedule (Pereira, et al., 2008). Regardless of the varied meanings of project success, most definitions include some concept of efficiency, productivity or the ratio of inputs used to the outputs produced and therefore productivity is generally considered to be an important component of project success.

Given that productivity is measured as the ratio of inputs to outputs, a comprehensive measure of productivity includes all the factors of production as inputs which are land, labour and capital (Parkin, 1990). While total productivity can be estimated using all the input factors of production, partial productivity can also be measured. This relates the outputs of a production process to each type of input separately, that is, each factor of production (Kudyba & Diwan, 2002). Labour only productivity measures are the most widely cited measure of economic efficiency although this is not the most comprehensive productivity measure. Despite the fact that a labour only productivity measure does not include all of the factors of production, metrics for software development generally only consider labour, as this is by far the main input to producing software (Boehm, 1987; C. Jones, 2000; Putnam & Myers, 2003).

As the main input to software development is labour, the motivation for organisations to improve productivity is compounded by the high cost of labour for roles involved in software development. As a result, any inefficiency in the production of software leads to significant extra cost for organisations. Software development is viewed as an essential activity with high value and this is evident in the high salaries and leadership roles entrusted to software engineers (Boehm, et al., 2007). Across all job categories in New Zealand, IT professionals get paid the highest (New Zealand Career Services, 2010; Trade Me, 2012). The average annual salary for those working in the computer system design sector (that is, software development) in 2011 was NZD $103,563 (New Zealand Ministry of Business Innovation & Employment, 2013). One of the key reasons for the high cost of software development staff is that there is a shortage of people with the required skills, both in New Zealand and globally. Six of the nine Information Technology professions with skill shortages in New Zealand involve the design and development of software systems (New Zealand Career Services, 2010). Furthermore, there are shortages of software development staff in a number of other countries (International Labour Organisation, 2008; Trauth, Quesenberry, & Huang, 2009; U.S. Department of Labor, 2012). Given the high cost of labour in software production, even small improvements in productivity can lead to major costs savings.

One driver for efficiency is the requirement to decrease the cost of software. In the 1990's economic conditions forced information system departments to focus simultaneously on decreasing costs while increasing software productivity (Mahmood, Pettingell, & Shaskevich, 1996). More recently analyses have found that cost reduction is still a major concern for software development organisations (Šmite, et al., 2010; Stober & Hansmann, 2009), in part due to the costs of large in-house software projects rapidly increasing (Mahmood, et al., 1996). Given this drive to decrease costs associated with software, improving software development productivity has become a focus of many organisations internationally (Zijiang & Paradi, 2009).

Despite major advances software producing organisations still face problems with productivity (Santos & Moura, 2009; Tan, et al., 2009) as many software development teams are unable to develop and maintain software cost effectively (Mens, et al., 2005). Using a productivity index based on lines of code per hour to analyse a database of over 6300 software projects completed between 1983 and 2003, Putnam and Myers (2003) show that high performing software development teams can be over 20 times more productive than low performing teams. Putnam and Myers investigate and discuss the factors that influence productivity but do not discuss any aspects of team diversity. Given the differences in productivity between teams, it is important for organisations to look for ways to improve the productivity of their low performing software development teams.

Factors Affecting Software Development Productivity

Factors affecting software development productivity have been identified in previous research across a number of disciplines. These factors each fall into one of three categories which are used here to describe the most critical software development productivity factors. These three categories are process, product and personnel factors.

Process factors

Process factors include the software development methodology used (Boehm & Turner, 2005; Dybå & Dingsøyr, 2008; Tan, et al., 2009), the effectiveness of the requirements capture approach (Cerpa & Verner, 2009; Chua & Verner, 2010; Hofmann & Lehner, 2001), the process maturity (Alyahya, Ahmad, & Lee, 2009; Gibson, et al., 2006), the degree of architecture and risk resolution undertaken, the use of software tools and the suitability of the documentation produced for supporting the software over its lifecycle (Boehm et al., 2000a).

A methodology is a process model which is required to provide predictability in order to measure and improve processes. “From the standpoint of a single organisation, it ought to be able to repeat whatever process it has.“ (Putnam & Myers, 2003, p. 36). Whether or not a formal methodology is used is one key process factor affecting software development productivity. Over the history of software development, a number of methodologies have been defined and used. The Standish Group (1995, 2001) observed that a formal and repeatable software development process would have resolved many of the factors that contributed to software development project failures. Failure for a software project is defined by The Standish Group as not achieving one or more of the three project success criteria “on time, on budget and with all features originally specified” (Standish Group, 2001, p. 1).

There has been debate regarding what type of software development process is most effective in improving productivity. Software development processes can be categorised as either “predictive” or “adaptive” characterising whether they are plan-driven or agile (Fowler, 2005). Predictive software methods are plan-driven and attempt to plan out a large part of the software process, consistent with traditional project management practices (for example, Project Management Institute, 2008). However, this can be a challenge in software development as not all activities can be predicted and the requirements of the software can change during development. In contrast, agile methods are designed to allow for change, and even encourage it (Fowler, 2005; Fraser, 2009; Stober & Hansmann, 2009). Larman (2004), a proponent of agile methods, describes the difference by stating “adaptive plans embrace change and opportunity; predictive plans fight or ignore it “ (p. 258). As the first key principle of agile development is to value "individuals and interactions over processes and tools" (Beck et al., 2001) the impact of any factors affecting the collaboration of team members is heightened. In line with this principle, agile development encourages face-to-face communication over documentation, email or phone based communication. This greater emphasis on face-to-face communications can alter team dynamics and how individuals interact with one another. Face-to-face communication in software teams tends to be ad-hoc and maximises the opportunities for information sharing at the time it is needed (Cherry & Robillard, 2009).

There are a number of formal agile methodologies such as Scrum, Lean software development, Extreme programming and Feature-driven development (Dybå & Dingsøyr, 2008). Scrum is one of the most formalised and widely adopted of the agile methodologies (Moe, et al., 2010). This methodology involves short delivery cycles called ‘sprints’ which are typically 30 days in duration. As with many agile methods, Scrum involves short daily stand-up meetings to discuss progress and roadblocks. Unlike some agile methods, a project manager is required, although the team is empowered to make many of the planning decisions itself.

With the adoption of agile development techniques, there has been an associated focus on the importance of team work (Stober & Hansmann, 2009). This is because, in the relative absence of development documents, agile methods rely on effective interpersonal relationships within the development team (Strode, Hope, Huff, & Link, 2011). It has been shown that agile practices improve communication within software development teams and with stakeholders external to the team (Pikkarainen, et al., 2008). These improvements in communication arise from greater opportunity for synchronous, face-to-face and informal discussion, and by facilitating direct communication between developers and the customer. This in turn leads to a higher level of productivity given that communication is more effective and less time is spent producing development documentation. The rise of agile software development has led to a greater emphasis on effective teamwork in software development (Šmite, et al., 2010; Tan, et al., 2009).

The key to an effective software project are clear, complete and accurate requirements that define what the system will do (Hofmann & Lehner, 2001). They specify what the software developers must produce. Changes to these requirements during the software development process introduce additional rework and effort, and negatively affect productivity (Boehm et al., 2000b; C. Jones, 2008; Osmundson, Michael, Machniak, & Grossman, 2003). While agile software development seeks to embrace change, there is a point at which requirements must be frozen to enable developers to create stable software (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003; Beck, 2000). The more effectively a project team can communicate with their customers the less often the requirements will change and the more productive a software team will typically be (Hofmann & Lehner, 2001). Effective communication facilitates requirements capture and thereby improves interactions between developers and customers. It enables developers to better understand what is required and thus are more likely to be able to deliver what the customer is requesting (T. Hall, Wilson, Rainer, & Jagielska, 2007; Zowghi & Bargi, 2011). As the programmers begin to build the software, clarifications are often needed and good communication aids this (Chua & Verner, 2010). Where those clarifications can be effectively communicated and understood by developers and customers, it is more likely the requirements will be met. Where communication is poor, it is often not until the software is used for the first time that it becomes apparent the requirements documented are not what was actually wanted and the software needs to be modified, or discarded and rewritten (Cerpa & Verner, 2009; Chua & Verner, 2010) leading to impaired productivity.

Product factors

Product factors refer to the software being produced and its accompanying artefacts (for example, user manuals and design documentation). These include the constraints and requirements placed upon the software product to be developed along with the technical factors relating to the hardware, the operating system and the platform used to develop the software. One of the constraints that arise from the platform on which the software will be deployed is the execution time constraint**.** The more a software solution needs to be developed in such a way as to minimise the system processing capability, the less productive the development of that software will be (Boehm, et al., 2000a). Another constraint that affects productivity is the platform volatility**.** The platform includes the hardware and system software (such as the operating system and database) the software product calls on to perform its tasks. If the platform only changes infrequently, such as once per year, then a project is likely to be more productive. If the platform changes frequently, such as every two weeks then a project is likely to be less productive (Boehm, et al., 2000a).

In addition to the platform, the product itself has constraints placed upon it that affect productivity. The first of these is the level of reliability required of the product which is the extent to which the software must perform its intended function (Boehm, et al., 2000a). If the effect of a software failure is only a minor problem then a higher level of productivity can generally be achieved during the development of the software. If a failure would cause harm to someone then productivity is negatively impacted. A second product related factor is the degree to which it is developed for reusability as greater effort is needed to develop software components that are to be reused in the future (Boehm, et al., 2000a). A third factor is the product's complexity (Boehm, et al., 2000a). If the software required is highly complex then more effort is required to build it and therefore productivity will be worse. Technical aspects of the product which affect productivity include the programming language used and the volatility of the underlying hardware and software used to develop the product (Boehm, et al., 2000a). The number of software instructions or lines of source code required to deliver end user functionality varies significantly between programming languages (C. Jones, 2008). Those languages that deliver more software with fewer lines of source code are more productive to develop with. These variations in the lines of source code are a major factor affecting overall productivity (DeMarco, 1995; P. Hill, 2010; Port & McArthur, 1999).

Personnel factors

Personnel factors affecting software development productivity are drawn from a wide range of disciplines, including management, project management, group theory, systems theory as well as software engineering. Factors identified as affecting productivity include team cohesion (Boehm, et al., 2000a) , staff availability (Foulds, Quaddus, & West, 2007; K. Maxwell & Forselius, 2000), staff turnover (Boehm, et al., 2000a) , staff capability and experience (Boehm, et al., 2000a; Foulds, et al., 2007; K. Maxwell & Forselius, 2000), and communication (Brodbeck, 2001; Daim et al., 2012; T. Hall, et al., 2007). In addition to these personnel-related factors typically associated with productivity, there is also literature on the significant and important role of the project manager in software development project outcomes. The following paragraphs focus on the most significant personnel factors, which are the use of teams, communication, project management factors, team cohesion and team composition.

The use of teams

Teams exist within the workplace to achieve common objectives. They can be more productive than the sum of the individuals but for a team to perform, its members must be both motivated and have the ability to achieve the set objectives (Earley & Mosakowski, 2000). Furthermore, good communication within a team enhances team performance (Kochan et al., 2003). Team composition affects communication within a team and therefore the composition of a team has a significant impact on its performance (Dahlin, Weingart, & Hinds, 2005; Stewart, 2006). As Wageman, Fisher and Hackan (2009, p. 194) say "Real teams (1) have clear boundaries; (2) are interdependent for some common purpose; and (3) have at least some stability of membership, which gives members time and opportunity to learn how to work together well".

An overriding trend in software development is that software is increasingly created by teams and not individuals (Lazear & Shaw, 2007; Yu, Bao, & Yang, 2009). The effectiveness of such teams is therefore of critical importance (Boehm, 1981; Gannon, 1979; Unger & Walker, 1977; Weinberg, 1998) and the main challenges on software projects relate to managing people and their interactions with one another (DeMarco & Lister, 1999). Tom DeMarco, as a member of a panel discussion, contends that "the success of a software project is likely to depend more on its sociology than its technology" (Fraser et al., 2007). One reason for these challenges is due to software's almost infinite malleability and the "tendency of software requirements to change" during the software production process (Cugola & Ghezzi, 1998, p. 103). Another difference is that the production line and the goods produced are intangible (Sharpe, 2001). Furthermore, software development is a knowledge intensive activity that requires the integration of knowledge which is dispersed across team members, thus heightening the need for effective collaboration within teams (Janz & Prasarnphanich, 2009). These characteristics increase the importance for software teams to have effective relationships (T. L. Lewis & Smith, 2008; Wong & Bhatti, 2009) and communication (Cherry & Robillard, 2009; Egan, et al., 2006).

A disadvantage of using teams for software development is that as development moves from work by a single person to team based, software development inefficiencies can occur (Fried, 1991; P. Hill, 2010; K. Maxwell, Van Wassenhove, & Dutta, 1996). The first inefficiency arises from the increased interpersonal communications required, because as each new team member is added, the paths for communication between members increases exponentially (Boehm, 1981; Ganssle, 2008). For example, when there are only two people involved there is only one communication path - that is between person A and person B. When a third person is added, the possible communication paths increase to three (A to B, B to C and A to C) and when a fourth person is added there are six potential communication paths (AB, AC, AD, BC, BD and CD). The second inefficiency arises from the additional work required when different developers need to integrate their work into a single software solution (Boehm, et al., 2000a). For example, where two developers are making different changes to the same program, they subsequently need to merge their changes which can be complex. Despite these potential "diseconomies of scale" (p. 30) the use of teams for software development is unavoidable given the size and complexity of software required to support today's society (Candrlic, Pavlic, & Poscic, 2006; Yu, et al., 2009).

The challenges arising from diseconomies of scale in software development teams have been sought to be addressed through the Team Software Process (TSP) developed by Watts Humprey (2000). This was developed to complement the Capability Maturity Model (discussed later in the subsection *Measuring and Improving Software Development Productivity*) on the premise that software as an engineering product generally requires effective teams to produce high quality software. The motivation for developing TSP was that software engineering teams "can do extraordinary work, but only if they are properly formed, suitably trained, staffed with skilled members and effectively led" (Humphrey, 2000, p. 1). A TSP study that involved analysis of project data and interviews with project participants in 20 software projects across 13 organisations found that teams using TSP delivered essentially defect-free software on schedule and with better productivity when compared with the productivity achieved before they started using TSP (Davis & Mullaney, 2003). However, TSP is closely aligned to the philosophy of the Capability Maturity Model whereby productivity can be improved through feedback using data captured throughout the process. For TSP to be implemented successfully an organisation must have already adopted the Capability Maturity Model (Yu, et al., 2009). Many organisations do not use the Capability Maturity Model, often because they have chosen an approach aligned with a different philosophy. Therefore, while TSP is useful to some organisations, there are many where it would not help. Furthermore, TSP is process centric and does not address interpersonal issues such as conflict and team composition other than in terms of team members' skills.

The inconclusive findings from previous research regarding whether larger software teams are more or less productive, and under what circumstances, gives rise to the need to examine team processes. Such an examination can help identify the dynamics between team members that can enable or inhibit productivity as larger teams are required to address complex software development challenges. A critical aspect of team work in software development is the communication that occurs, as this underpins the activies required to produce and deliver software.

Communication

The types of communication which can affect software development productivity can be categorised using systems theory (Almaney, 1974). With the software development team being the system under examination, some communications are within the system while others relate to "boundary spanning" exchanges (Modaff, DeWine, & Butler, 2008, p. 77) with stakeholders and other teams. Communication in software projects has previously been analysed using the Sender, Message, Channel, Receiver model developed by Berlo (1960) to identify where there are significant issues affecting productivity. The channel selected for communication in software projects has been found to affect productivity (Daim, et al., 2012; T. Hall, et al., 2007). Face-to-face enables richer and more complex communication to occur as it is synchronous as opposed to email which is asynchronous (Modaff, et al., 2008). Synchronous transactional communication makes greater use of feedback loops and is therefore more complex (Miller, 2004; Sligo & McLean, 2000) which allows richer, non-standardised and more effective transmission of information (Brodbeck, 2001). Transactional communication can be contrasted with one-way informational transmission which is how communication has traditionally been modelled (Craig, 1999). This one-way type of communication is generally not considered useful for most situations on software projects as some form of feedback is almost always required (de Souza, Quirk, Trainer, & Redmiles, 2007; Fuks, Raposo, Gerosa, & Lucena, 2005; Santos & Moura, 2009). The communication network required for a software project has many nodes (that is, people and systems who need to send or receive communication) and paths between those nodes (Boehm, 1981; Ganssle, 2008). Furthermore, the communication is often multidimensional, involving different types of information and utilising a variety of channels (T. Hall, et al., 2007; Marttiin, Lehto, & Nyman, 2002). For these reasons, software projects typically require complex communication and this can often present challenges (Hayes, 2003).

One approach to deal with complex communication required on software development projects has been the development and evolution of collaboration tools that support and enhance team interactions in software development (Candrlic, et al., 2006). These tools include software configuration management tools, application lifecycle management tools and work group collaboration. Employing these tools helps to address communication when direct interpersonal communication is not possible, or practical, such as in distributed or global software development (Daim, et al., 2012; Herbsleb, 2007; Šmite, et al., 2010). However tools such as these do not address underlying sources of conflict or communication challenges which would exist even if face-to-face interpersonal communication occurred (Barki & Jon, 2001; Goyal, Maruping, & Robert, 2008; Holmes & Marra, 2010).

Project management factors

The function of a project manager.

Project managers play a significant role in software projects and therefore influence team performance (Ehsan, et al., 2010; Thite, 1999). They perform a critical role as they act as a channel for disseminating information to both stakeholders and team members (Gillard & Johansen, 2004). Team composition is often determined, or at least influenced by project managers (Sebt, et al., 2010) and they have a good understanding of the factors that affect the productivity of a project (Ehsan, et al., 2010; Wang, 2009). For these reasons, the leadership styles and characteristics of project managers have a major influence on how their projects perform.

Project management differs to other types of management as project teams are often staffed with a range of diverse people with varying skills. This contrasts with functional teams where those in the team are generally drawn from the same discipline or trade (Hartley, 2008; Pinto, 2009). Furthermore, project teams typically only work together for the duration of the project and often have not worked together prior to the project (Anantatmula, 2010; Cleland & Ireland, 2006). This gives rise to the need for techniques and leadership approaches unique to project management (Parker, Craig, & Craig, 2008). Project managers need to quickly build cohesiveness within the team, reducing any barriers that exist between team members (Gehring, 2007) and resolving uncertainty inherent in the early stages of projects (Atkinson, Crawford, & Ward, 2006).

Project managers have their biases and blind spots. They often tend to be optimistic in their assessment of the state of their project (Snow & Keil, 2002) which is considered to be “role induced bias” (Heemstra, Kusters, & de Man, 2003, p. 1). Individuals interpret information differently based on their situation and in the case of project managers, pressures to report successful results often leads to an overly optimistic interpretation of risks as they are accountable for project outcomes (Siemiatycki, 2008). These project management characteristics all have the potential to impact on a project's productivity.

Leadership.

Project managers perform a range of functions such as planning, organising and controlling (Anantatmula, 2010) but leadership is frequently singled out as an important activity (Nixon, Harrington, & Parker, 2012; Ong, Richardson, Yanqing, Qile, & Johnson, 2009; Yang, Huang, &

Wu, 2011). Alternative terms which are closely related to project leadership are managing personnel factors (Belout & Gauvreau, 2004) or managing human resources (Dong, Li, Li, Yang, & Wang, 2009; Thomas & Mullaly, 2007). Although these terms are not interchangeable, they all relate to the way in which the people, including the team are managed to produce project success. Leadership is considered one of the soft-skills which project managers require, along with other soft skills such as communication, negotiation and professionalism (Skulmoski & Hartman, 2010). While it is generally accepted that leadership or human resource management by the project manager plays a role in the success of projects, just how significant that role is, is widely debated (Anantatmula, 2010; Belout & Gauvreau, 2004; Nixon, et al., 2012; Pinto & Prescott, 1988).

The leadership style employed by a project manager also affects project outcomes. Some have a more dictatorial leadership style (Bruce & Langdon, 2000; Sukhoo, Barnard, Eloff, & Van Der Poll, 2005) while others are more participative or consultative (Bass, Valenzi, Farrow, & Solomon, 1975). One approach for categorising different project management styles is to consider whether they are focussed on the tasks or the people (J. R. Turner & Müller, 2005). Task-focussed, or transactional project management styles tend to be more concerned about completing the agreed tasks whereas people focussed styles are more concerned about stakeholder relationships (Holmes & Marra, 2010). Different styles of project management have been found to be more effective with certain types of projects (Dulewicz & Higgs, 2005; Müller & Turner, 2007) and with certain project stages (Frame, 2003; J.R. Turner, 2008). Transformational leadership which focuses on people can be better on more complex projects while project management using transactional leadership or a task focus can work best on simpler projects (Dulewicz & Higgs, 2005). A project manager's characteristics such as personality (Wang, 2009), soft skills (Skulmoski & Hartman, 2010) and emotional intelligence (Clarke, 2010) can also significantly affect team productivity.

Trust.

A key consideration of project leadership is to create a team environment which fosters effective working relationships. Such relationships within a team have been found to be underpinned by trust, and this is an area in which the project manager makes a significant difference (Anantatmula, 2010; Ochieng & Price, 2010). To increase trust within a team, a project manager can reward honest and open communication, as well as making use of team building activities (Ochieng & Price, 2010). A quantitative study of project management practices in 550 organisations internationally found that “establishing an environment of trust” (Anantatmula, 2010, p. 15) is one of the most important drivers of project management success. In software teams, low trust leads to poor relationships which negatively impact on team performance as good and open team relationships are crucial for successful software development (Wong & Bhatti, 2009).

Team cohesion

Team cohesion is a productivity factor in software development (Boehm, et al., 2000b; Foulds & West, 2007) and team performance in general (Anderson & West, 1998; Burningham & West, 1995; van Knippenberg, Dawson, West, & Homan, 2011). It is defined in the COCOMO II model by Boehm et al, as:

> The Team Cohesion scale factor accounts for the sources of project turbulence and entropy because of difficulties in synchronizing the project’s stakeholders: users, customers, developers, maintainers, interfacers, others. These difficulties may arise from differences in stakeholder objectives and cultures; difficulties in reconciling objectives; and stakeholders' lack of experience and familiarity in operating as a team. (2000a, p. 20)

Boehm et al, go on to list the following four characteristics which determine overall team cohesion: “1) consistency of stakeholder objectives and cultures; 2) ability and willingness of stakeholders to accommodate other stakeholders' objectives; 3) experience of stakeholders in operating as a team; 4) stakeholder team building to achieve shared vision and commitments" (2000a, p. 20). The first of the four characteristics is highlighted in a number of studies which state the importance of shared goals in achieving team cohesion (Anderson & West, 1998; Burningham & West, 1995; Sherif, 1966; van Knippenberg, et al., 2011). Over the course of a software project, the degree of team cohesion changes (Snowdeal-Carden, 2013). As the team relationships evolve, different challenges are encountered and overcome, and as different staff join and leave the project the amount of cohesion varies (Wellington, Briggs, & Girard, 2005).

The strength of the relationships amongst project members is important for team cohesion. Team relationships can be examined using social exchange theory (SET) which explains how relationships occur and evolve (Cropanzano & Mitchell, 2005). SET is derived from classical exchange theory which deals with the dynamics and basis on which exchange occurs (Molm, 2003). According to SET, relationships are underpinned by reciprocity and negotiated rules (Blau, 1964). Exchange can occur in response to a variety of motivations, for example economic exchange may occur for profit and typically involves contracts in some form. In contrast, social exchange involves reciprocity and negotiation. Reciprocity relates to the exchange of (sometimes unstated) benefits while negotiation involves the people who agree to the terms of the relationship (Mach, Dolan, & Tzafrir, 2010). Considering these basic tenets of social exchange, managers can improve team cohesion through team building activities (C. Klein et al., 2009) and creating an inclusive, transparent team environment (Whitworth & Biddle, 2007). Such measures provide catalysts for negotiating and agreeing the terms of team relationships.

Team composition

The composition of a team is an important factor in determining the degree of cohesion and therefore overall performance. A mix of complementary skills, experience and background enables a team to perform at a high level. "Cognitive resource diversity theory" (Horwitz, 2005, p. 224) underpins the argument that diversity benefits a team as there is a greater pool of experiences, skills and perspectives to draw upon (Hambrick, Cho, & Chen, 1996; Post, et al., 2009). Teams which are ethnically diverse are more likely to have the variety of experiences, skills and perspectives which provide a greater pool of cognitive resources that leads to higher performance (Jehn & Bezrukova, 2004; P. Richardson, 2005; Winkler & Bouncken, 2009). However, intervening processes such as collaborative planning (Woolley, et al., 2008) or enhanced communication (Horwitz, 2005; Nam, Lyons, Hwang, & Kim, 2009) may be required for them to contribute to performance benefits. Other aspects of team composition which have been examined in relation to team performance are personality type (Bielefeldt, 2009; Chatenier, et al., 2009), demographic attributes (Chowdhury, 2005; Cunningham, 2007; Pelled, 1996) and team size (Fried, 1991; P. Hill, 2010; K. Maxwell, et al., 1996).

Personality type indicators have been used to analyse the effect of composition on the productivity of software development teams. A range of personality types in software teams has been found to improve software development productivity through increased creativity, innovation and flexibility and the productivity gains were greatest when more time was given to enable relationships to form (T. L. Lewis & Smith, 2008). Where software teams had an uneven mix of personality types (a strong weighting towards one personality type for example) majority and minority views were formed which in turn led to conflict. Myers-Briggs Type Indicator personality typing has been used to select team members with personality profiles that match the role they are to perform, for example, programmer, designer or project manager, to improve the success of software projects (Capretz & Ahmed, 2010).

Where a team is split into two relatively homogeneous groups, team cohesion can be negatively affected (Bezrukova, Jehn, Zanutto, & Thatcher, 2009; Lau & Murnighan, 1998). This is referred to as fault lines and one example of this which can occur in software development teams is between permanent and contract staff (Ang & Slaughter, 2001). However, if people from different groups get to know each other early in the group formation process, fault lines are less likely to form. One important aspect of team composition is the diversity of a team (Pelled, 1996; L. Turner, 2009) which is discussed in detail in the section *Ethnic Diversity*.

Diversity and Team Performance

Workplace Diversity

Diversity in the workplace describes differences between workers including aspects such as gender, ethnicity, age, religion, sexual orientation and 3 (Canas & Sondak, 2008). Managing workplace diversity involves “establishing a heterogeneous workforce to perform to its potential in an equitable work environment where no member or group of members has an advantage or disadvantage” (Cascio, 2003, p. 121). The term 'heterogeneous' can be problematic in this context as some research (for example, Carayannis, Kaloudis, & Mariussen, 2008; Schumpeter, 1911) into heterogeneity in the workforce considers a wider meaning of diversity. These analyses use a broader definition of heterogeneity, such as differing perspectives on problem solving. In order to provide a common taxonomy and understanding when discussing workplace diversity, models of diversity have been developed.

One such model adopted for analysing diversity is defined by Loden and Rosener (1991) and includes primary and secondary dimensions. The primary dimensions of diversity are age, ethnicity, gender, mental / physical abilities, race and sexual orientation. The secondary dimensions are “the mutable differences that we acquire, discard, and/or modify throughout our lives” (Loden & Rosener, 1991, p. 19). These include work style, education and present location. If diversity is differences in people, then inherent in that are differences in background, interpersonal style, work style and communication style (Capezio, 1998). These differences can give rise to challenges in achieving effective teams that operate productively. Diversity in organisations requires managers to be able to deal with conflicting perspectives while being able to maintain “integrity and fair-mindedness and a persuasive, congenial personality” (Bureau of Labor Statistics, 2007).

The need for workplace diversity can be argued based on a business case or a call for social justice (Glover & Evans, 2011). On one hand, it can be argued that society would be fairer and more just if the workplace better represented a cross-section of the population. On the other hand, the business case for diversity in the workplace is based on reducing costs, attracting skilled staff and growing the business (Canas & Sondak, 2008). Using the business case approach, one aspect of reducing costs relates to legal proceedings that arise from actions which prevent diversity in the workplace and can be a significant cost to the organisation. For example, in 2003, the clothing retailer Abercrombie & Fitch was sued for racial discrimination against Latino and Asian workers. The company were required to pay $50 million to settle the racial discrimination case (United States Equal Employment Opportunity Commission, 18 November 2004). In order to increase ethnic diversity in the workplace it is necessary to avoid such discrimination and therefore supporting workplace diversity can help reduce costs by reducing the risk of legal action. Business growth can arise from organisations expanding to overseas markets. Having culturally diverse teams better enables organisations to understand the cultures of potential markets (P. Richardson, 2005). Business growth can also arise from better problem solving capabilities by teams. as culturally diverse teams (with culture being an important aspect of ethnicity) have been found to be better at problem solving (Watson, et al., 1993). (This is discussed further in the section that follows entitled *Ethnic Diversity*). Finally, attracting and retaining scarce skills may be a sufficient business case on its own for workplace diversity in industries with skills shortages.

Team Diversity

Teams exist in many contexts, such as sporting and community groups, but the examination of team diversity primarily focuses on teams as work groups within organisations (Horwitz & Horwitz, 2007; Jackson, Joshi, & Erhardt, 2003). Each member of a team possesses characteristics which can be either objectively or subjectively identified (van Knippenberg & Schippers, 2007). The diversity of the team represents the degree to which these characteristics differ between team members and how these differences affect how the whole team performs, rather than how the diversity affects the performance of individuals (van Knippenberg, 2007).

The form which diversity takes within a team is important. Harrison and Klein (2007) developed three topologies of diversity within a team for a specified attribute, such as ethnicity. These topologies are separation (two polarised groups), variety (an evenly spread range) and disparity (one individual being significantly different from the rest of the team). Diversity fault line theory extends the separation topology and explores the effect (generally negative) of two polarised groups within a team (Lau & Murnighan, 1998; van Knippenberg, et al., 2011). Fault lines can arise in teams based on ethnic groupings (Lau & Murnighan, 1998; Leong & Ward, 2000). When fault lines form along demographic lines such as ethnicity, this can lead to the "formation of sub-groups and the emergence of inefficient in-fighting" (Lau & Murnighan, 1998, p. 337).

The effect of diversity in teams has been the subject of many studies over the last 20 years (Harrison & Klein, 2007) and there is strong evidence that diversity affects how a team operates (Horii, Jin, & Levitt, 2005; Horwitz, 2005; Jehn & Bezrukova, 2004; van Knippenberg, 2007; Winkler & Bouncken, 2009). The results of such studies are nuanced and varied, with many finding that team diversity has both positive and negative effects. Van Knippenberg explains that a diverse team is more likely to have a "broader range of task-relevant knowledge, skills and attributes" (2007, p. 13), and because of the "need to integrate and reconcile diverse perspectives" (2007, p. 13), this is more likely to stimulate creativity and innovation. However, relationship aspects can have a negative effect in diverse teams because, based on a social categorisation perspective (K. Y. Williams & O'Reilly, 1998), people see others who are similar to them as being part of their own group and those who are dissimilar as part of a different group. "The social categorisation perspective predicts that team diversity disrupts group processes because group members are less prone to like, trust and co-operate with dissimilar others" (van Knippenberg, 2007, p. 12). This suggests that team cohesion is affected by team diversity and is therefore likely to influence productivity in software development teams.

It is evident that diversity affects how teams operate but the relationship between this and team performance is not simple or direct. For this reason it is necessary to develop a nuanced understanding of the nature of team diversity (Mannix & Neale, 2005), and in particular an understanding of the role of “mediating processes and moderating variables” (van Knippenberg & Schippers, 2007, p. 15) that affect how diversity influences team performance. A mediating process is a mechanism which generates a relationship between two variables such as team diversity and team performance, while a moderating variable alters an existing relationship between two variables (Baron & Kenny, 1986). One example of a factor which alters the influence of team diversity is where team members have shared objectives as this enhances the benefits of team diversity (van Knippenberg, et al., 2011). Another example is that team perceptions regarding the relevance of social categorisation can determine how team diversity affects performance (Meyer, Shemla, & Schermuly, 2011). Identifying such factors enables a better understanding of how team diversity can improve innovation and problem solving (Brandes, et al., 2009; Joshi & Roh, 2009; van Knippenberg & Schippers, 2007).

The Organisational Context of Diversity

The effects of diversity in teams are largely dependent on the organisational context (Jehn, 1995; Kochan, et al., 2003; K. Y. Williams & O'Reilly, 1998). Where an organisation acknowledges and actively manages diversity, then adverse effects of team diversity such as miscommunication, conflict and lack of team cohesion are more likely to be mitigated (K. Y. Williams & O'Reilly, 1998). A model of the effect of team diversity on outcomes is presented by Kochan et al. (2003) and shown in Figure 1. This diagram shows that critical aspects of the organisational context determine whether diversity has a positive or negative effect within the organisation. Furthermore, the effects of diversity occur via group or team processes, which also determine whether the effects of diversity will be positive or negative.

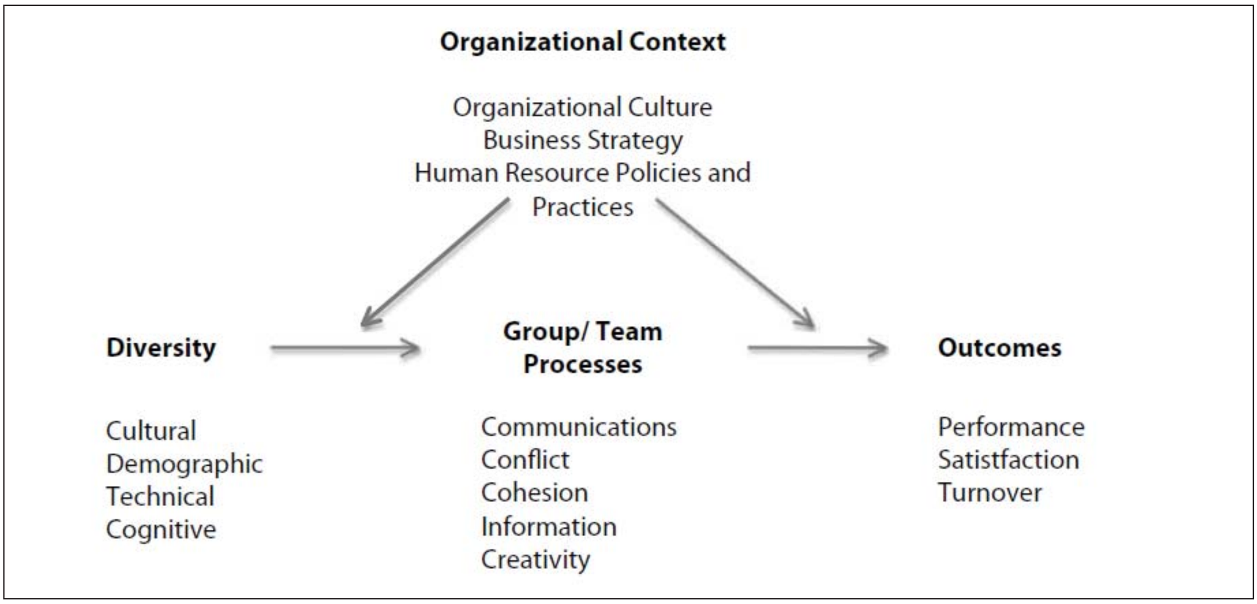


Figure 1. Effects of diversity on group processes and outcomes (Kochan, et al., 2003)

An organisation's culture, systems and processes for managing diversity within teams significantly affects how diversity influences team performance (Joshi & Roh, 2009; Kochan, et al., 2003; Richard, Kochan, & McMillan-Capehart, 2002; Walsham, 2002). Organisations' endeavours to manage diversity typically involve training to improve individuals' awareness and knowledge of diversity in the workplace (Curtis & Dreachslin, 2008). When members of a group believe diversity is positive, the group is likely to perform better (Van Dick, Van Knippenberg, Hagele, Guillaume, & Brodbeck, 2008). Despite this finding, the link between improving an individual's awareness of diversity and the individual changing their behaviour is weak (Curtis & Dreachslin, 2008). Most studies of interventions in the workplace to improve diversity focus on the attitudes (Sawyerr, Strauss, & Yan, 2005), beliefs (Van Dick, et al., 2008), perspectives (Ely & Thomas, 2001) and preferences (Paulus, Nakui, Parthasarathy, & Baruah, 2004) of individual employees or team members towards diversity.

One way in which organisations and teams can alter the effects of diversity is through awareness and understanding of different communication styles required for effective communication within diverse teams. Without an awareness of nuances in language and differences in style, the potential for miscommunication is significant when interacting with people who are of different ethnicity (Loden & Rosener, 1991). Communication underpins all business operations but can also be a source of misunderstanding and workplace entropy. The importance of communication to share information is amplified as a business's dependence on high quality information increases (Sligo, Fountaine, O'Neill, & Sayers, 2000). An important difference in the way organisations communicate and operate is determined by the purpose of the organisation. As the goals of government organisations differ to those in the private sector, the sector of the organisation affects how it operates, and therefore how it deals with diversity.

Public and Private Sector Differences

The operations, systems and processes of government organisations are significantly different to private organisations (Campbell, McDonald, & Sethibe, 2010), particularly in IT (Sethibe, Campbell, & McDonald, 2007; Vilvovsky, 2008). These differences affect risk taking (Bozeman & Kingsley, 1998), innovation (Bhatta, 2003) and leadership (Fernandez, Cho, & Perry, 2010). These organisational differences have implications for the way team diversity is managed in the public sector when compared to the private sector, and how it affects productivity.

In order to compare and contrast the private and public sector, it is necessary to provide a clear classification of organisations. Definitions distinguishing the government or public sector from the private sector vary (Campbell, et al., 2010; Rainey, Backoff, & Levine, 1976). A broad definition for government is the organisations that provide goods and services on behalf of the government (Sethibe, et al., 2007). The non-government sector includes predominantly the private sector but also includes the voluntary, not-for-profit, or "third sector" (Hull, Gibbon, & Branzei, 2011, p. 3). The purpose and goals differ significantly between the sectors with private sector organisations typically being profit driven and existing in a competitive environment. In contrast, government organisations have their goals set for them by parliament and generally operate as a monopoly with no direct competition.

There are critical differences between public and private organisations which can affect how they operate. One is that the government sector's priorities and goals can change each electoral term, altering the resources available to government projects, including IT projects (Campbell, et al., 2010). As a result, the direction and focus of IT projects can change part way through a project, which can alter the way in which they are managed. Furthermore, it can be difficult for public sector managers to justify approaches considered to be risky even if there are significant potential rewards such as faster and cheaper delivery of software (Rocheleau & Wu, 2002). This can be because public sector managers have many competing objectives to meet (Meynhardt & Diefenbach, 2012) or because rules, procedures and "red tape" (Bozeman & Feeney, 2011, p. 1) can prevent risk taking (Feeney, 2012). It has also been argued that managers in government organisations are subject to greater scrutiny than their private sector counterparts as there is greater accountability imposed on government agencies (Nicoll, 2005). Managers of IT projects in the public sector face additional risks over and above their counterparts in the private sector arising from one year budgets, multiple stakeholders with competing goals, highly regulated procurement and extreme risk aversion (Dawes et al., 2004).

Comparing government and private sector productivity is difficult as government and private sector outputs are often different (Bel, Fageda, & Warner, 2010; Phelps, 2010; H. Simpson, 2009). The outsourcing of government software development projects is one way to attempt to compare the productivity of the two sectors. However outsourcing in government generally has wider implications for public sector outcomes and it is therefore not possible to separate and compare productivity rates (Dibbern, Goles, Hirschheim, & Jayatilaka, 2004; Ravishankar & Navneet, 2006). Furthermore, government agencies do not generally measure software development productivity because they do not operate in a competitive environment (C. Jones, 2008).

Despite these differences between public and private organisations, sector differences have been largely ignored in previous analyses of information systems (IS) performance (Chan & Reich 2007). There have been some studies of IS projects in the public sector (for example, Bretschneider, 1990; Gauld, 2007; Rocheleau & Wu, 2002) but few have focussed on productivity differences between software development in the public and private sector. An analysis of over 5000 software projects in the ISBSG database (International Software Benchmarking Standards Group, 2009) showed that government software projects are notably less productive than private sector projects (Congalton, 2011). However, the causes of the difference in software development productivity between the public and private sectors do not appear have not been investigated in any research to date.

Diversity in Software Development

In order to review previous research on diversity in software development teams, a series of leading article databases were searched using a variety of search terms. These searches were repeated at regular intervals from 2010 through to 2013 and the results included both journal articles and conference proceedings. Alerts were set up on key databases using search terms so new relevant articles could be considered as they were published. Examples of the search terms and article databases used are listed in Table 2. These are not the only search terms used for the literature review, but are the key terms used to find previous work on the influence of ethnic diversity on software development productivity.

Table 2. Search terms and article databases for previous work

Example Search Terms Used	Article Databases
(Diversity OR Culture OR Cultural OR Ethnic) AND ("Software development" OR "Software engineering" OR "Information systems" OR "Information technology" OR "Software production" OR "Systems development") AND (Productivity OR Performance OR Effectiveness OR Outcome OR Output)	IEEE Xplore ACM Digital Library SpringerLINK Scopus Web of Science JSTOR Business Source Complete Google Scholar

The literature search revealed that diversity in software development teams has been shown to have various effects on the outcomes of those teams. Table 3 lists existing studies into the effect of diversity in software development teams. This table shows varying aspects of diversity that include responses to conflict, personality type and cultural differences in temporal perceptions.

Table 3. Studies into the effect of diversity in software development teams

Study	Method	Sample	Aspect of diversity	Findings
(Barki & Jon, 2001)	Quan	265 IS staff and 272 users working on 162 software development projects	Response to conflict and conflict management style	The impact of interpersonal conflict was perceived to be negative, regardless of how it was managed or resolved.
(Pieterse, et al., 2006)	Quan	82 software engineering university students	Personality type using Keirsey-Bates Temperament Sorter	Personality diversity is a strong predictor of success especially during the initial phases of team growth.
(Liang, et al., 2007)	Quan	85 team members from 16 software development projects	Knowledge diversity (KD) and value diversity (VD)	KD increases task conflict, which in turn has positive effects on team performance. VD increases relationship conflict, which in turn negatively affects team performance.
(T. L. Lewis & Smith, 2008)	Quan	38 software engineering university students	Personality type using Myers Briggs Type Indicator	Team composition was an important determinant of outcomes and teams dominated by problem solving type personalities led to negative outcomes.
(Shachaf, 2008)	Qual	41 team members from nine countries employed by a Fortune 500 company	Cultural diversity	Differences in language, verbal styles, and nonverbal styles negatively influenced team effectiveness, but this was mitigated through use of appropriate ICT communication tools.
(Egan et al., 2009)	Quan	200 software testers in a Fortune 500 company	Cultural differences in temporal perceptions[1]	Cultural differences impair communications.
(Liang, et al., 2009)	Quan	299 members of 75 software development teams	Informational diversity[2]	Informational diversity leads to improved software quality if learning opportunities are maximised.

Some of the previous research on diversity in software teams provides frameworks for analysing diversity (Dafoulas & Macaulay, 2001; Walsham, 2002) and others report on research undertaken into team diversity, focusing on aspects other than ethnicity. Cultural diversity is an

[1] Temporal perceptions affect how individuals view time and its significance.
[2] Informational diversity refers to differences in knowledge and perspectives, arising from factors such as skills, experience and education (Liang, Jiang, Klein, & Liu, 2010)

aspect of ethnicity, and while this has been studied in software teams (Egan, et al., 2006; Shachaf, 2008) this has not been combined with an examination of the effect of this diversity on software development productivity. The following section presents and discusses literature relating to the concepts of ethnic diversity.

Ethnic Diversity

Definitions of ethnicity

Ethnicity encompasses both biological race and culture (A. Smith, 1986). Ethnicity has its origins in "biological race" (Hunley, Healy, & Long, 2009, p. 35) but ethnicity has also been described as a social construction (Cormack, 2010; Norval, 2004). The race of an individual is determined by their ancestry which cannot be changed during their life. By contrast, the aspect of ethnicity which is socially constructed depends on a society's perceptions of itself and of other groups of people (Cormack, 2010). Culture is generally considered socially constructed and is an important component of ethnicity as this includes values and behaviours, and affects how individuals interact with others (E. T. Hall, 1983; Winkler & Bouncken, 2009). While ethnicity is self identified, it is possible for someone to know, with a relatively high degree of accuracy, the ethnicity of others they know well (Statistics New Zealand, 2005b; Zsembik, 1994).

Some information systems literature has focused on national culture to capture differences in values and beliefs (Ali & Brooks, 2008; Earley & Mosakowski, 2000; Hofstede, 1980a). Such literature associates culture primarily with a nation, but many nations contain significantly distinct cultural groups and cultures often exist across multiple nations (Eriksen, 2003; Sanders, 2002). National boundaries are also relatively arbitrary, often being the result of wars or political expediency (A. Smith, 1986). Furthermore, in the increasingly globalised society with access to international information through the internet, national boundaries are becoming less relevant

(Castells, 2009). Therefore, it has been argued that ethnicity is a better indicator of values and beliefs as this transcends national boundaries (Eriksen, 2003; Myers & Tan, 2002; A. Smith, 1986).

Ethnicity is defined as “having common racial, cultural, religious, or linguistic characteristics” (Oxford English Dictionary, 2013) but has also been used as a demographic classifier for health studies and in other areas. Ethnicity is represented as a variable in epidemiological research to describe health data, but has led to unintended interpretations (Senior & Bhopal, 1994). Ethnicity gathered in an epidemiological context often focuses on physiological and biological trends, whereas ethnicity in social science is more often used to examine values, beliefs and traditions (Airhihenbuwa, 2007; Ford & Harawa, 2010). If the data is then used for a purpose other than what it was collected for, invalid conclusions may be drawn (Bhopal, 2006).

There are a multitude of indexes and classification systems measuring ethnicity and as each classification generally has a specific purpose, the definition of ethnicity is often purpose specific and therefore context dependent. This has led to attempts to standardise classification systems for ethnicity (Smart, Tutton, Martin, Ellison, & Ashcroft, 2008). A hierarchical classification system of ethnic categories is used by Statistics New Zealand (2005a) as a basis for reporting a wide range of national data. These standards are based on equivalent Australian standards which in turn refer to previous work on defining ethnicity (A. Smith, 1986).

Ethnicity in New Zealand

Organisations in New Zealand employ a wide variety of ethnicities and in May 2013 there were over 200 different cultures in the Auckland workforce (Court, 2013). Employing and managing people of different cultures requires some understanding that work practices, cultural values and communication norms will differ between each ethnicity (Tiatia, 2008). As a result, the New Zealand government funds training and awareness programmes to increase employers' understanding and appreciation of the benefits of cultural diversity in the workplace (Cassidy-MacKenzie, 2013a). Despite this, in recent years the number of complaints about racism and discrimination in the workplace have increased (EEO Trust, 2013). Some of these complaints relate to racial harassment and others for discrimination such as job advertisements which specify the applicant must have English as a first language (Cassidy-MacKenzie, 2013b).

Research into ethnicity in New Zealand has mainly focussed on the identity of Maori, the indigenous people (Harding, Sibley, & Robertson, 2011; Kukutai, 2007; Sibley, et al., 2008). However, there is also a growing body of research into ethnicity related issues in New Zealand for significant immigrant populations such as people from the Pacific region (R. S. Hill, 2010) and from further afield (Lewin et al., 2011). An important component of the immigrant population in New Zealand is Asian with people of Asian ethnicity making up 9.2% of the total population (Badkar & Tuya, 2010). The Asian workforce in New Zealand are generally highly qualified and are projected to be an increasing component of the New Zealand workforce (Badkar & Tuya, 2010). Despite the Asian contribution to the New Zealand workforce, there is a lack of Asian people in senior management of New Zealand organisations (M. Chen, 2013). Within New Zealand there have been calls by researchers involved in cross-cultural studies to support immigrant populations having greater participation in New Zealand society (Ward & Liu, 2012). Although New Zealand's immigration policy seeks qualified migrants, when they move to New Zealand, they are generally under-employed. That is, the qualifications of many immigrants are not utilised as they are unable to secure employment in the areas in which they are qualified (Ward & Liu, 2012).

Individuals may define their own ethnicity. For instance, ethnicity may change over time (ethnic mobility) and in different contexts (Callister, Didham, & Kivi, 2009; Sanders, 2002; Statistics

New Zealand, 2005b). Ethnic mobility is relatively high in New Zealand, particularly for Maori, Pacific Island and Asian individuals (Carter, Hayward, Blakely, & Shaw, 2009). People sometimes change the ethnic group they self-identify with when they move countries (Sanders, 2002; L. Simpson & Akinwale, 2007). When a child grows up they may define their ethnicity for themselves, rather than have it defined by their parents (Carter, et al., 2009). Changes in the way an ethnicity is perceived by society may also lead to an individual changing the ethnicity with which they identify.

The previous section presented a discussion of the different meanings of ethnicity and argued that ethnicity is a better indicator of values and beliefs than national culture. However cultural values are integral to ethnicity and in the following section a review of cultural values is presented, followed by key factors that can arise as result within ethnically diverse teams.

Cultural values

Ethnic differences are important as they represent differences in cultural values. These differences have been modelled as generalised dimensions that account for some of the effects of ethnic diversity. Four critical dimensions reflecting cultural values were originally proposed by Hofstede (1980a) and further dimensions have since been added by the GLOBE project (House, Javidan, Hanges, & Dorfman, 2002), Bond (Bond, 1988; Fang, 2003), G. H. Hofstede, G. J Hofstede and Minkov (2010), and other groups of researchers. The original four dimensions proposed by Hofstede have been widely discussed and adopted in research into cross-cultural issues. Hofstede labels these four dimensions "Power Distance, Uncertainty Avoidance, Individualism-Collectivism, and Masculinity-Femininity" (Hofstede, 1980b, p. 43). Power-distance is the degree to which people accept unevenly distributed power. Countries such as Malaysia and the Philippines have a very high power-distance rating as their cultural values are such that they accept large disparities in power. By contrast, New Zealand has a low power-distance ranking as large disparities in power are less acceptable. Uncertainty avoidance represents the way in which different cultural values affect the tolerance individuals have to uncertainty when deciding what actions to take. Masculinity-Femininity is based on gender stereotypes whereby a masculine society is characterised as assertive, tough and materialistic, and a feminine society is more interested in caring, feelings and quality of life. The fourth of Hofstede's original cultural dimensions, Individualism-Collectivism, contrasts cultures where individuals take care of themselves (individualism) with those where people are strongly loyal to a wider group (collectivism). The Individualism-Collectivism dimension is considered by some to be the most salient cultural dimension (J. A. Lee, 2000; Li & Aksoy, 2007; Triandis, 2004) and there has been debate regarding whether these are two ends of a single dimension, or whether they are in fact two related but independent dimensions (Chirkov, Ryan, & Willness, 2005; Li & Aksoy, 2007; Wagner Iii, 1995). Teams where members have collectivism as a cultural value have been found to perform better than those where team members are more individualistic (McAtavey & Nikolovska, 2010).

Criticism of Hofstede's dimensions of cultural values have included biases in the sampling method used as all of the subjects were from IBM (M. L. Jones, 2007), shortcomings in focussing only on the values of culture to the exclusion of other aspects (Taras, Kirkman, & Steel, 2010) and the use of geography as a basis for difference (Chao & Moon, 2005; Myers & Tan, 2002). The data used by Hofstede is also relatively old having been gathered prior to 1980 and may not accurately reflect today's societies. However, this criticism has been refuted on the basis that cultural values do not change significantly over time (Minkov & Hofstede, 2011) and this view is supported by empirical analysis of western cultures over three decades (Inglehart, 2006). Other notable approaches to framing differences in cultural values include work by Trompenaars (Trompenaars & Hampden-

Turner, 2004), GLOBE's cultural dimensions (Hofstede, 2006; House, et al., 2002) and structuration theory (Walsham, 2002). Despite shortcomings with Hofstede's dimensions of cultural values, they are widely cited and used in studies of the effect of culture differences (M. L. Jones, 2007; Taras, et al., 2010; Triandis, 2004).

Innovation

As ethnicity reflects critical aspects of an individual’s values, beliefs and therefore behaviours, ethnic diversity has been found to influence team performance (Brandes, et al., 2009; P. Richardson, 2005; Watson, et al., 1993; Winkler & Bouncken, 2009). In their qualitative study involving 105 interviews in five innovation teams, Winkler and Bouncken (2009) found that ethnic diversity in the workplace can aid in innovative activities such as during the feasibility stage of a project as different viewpoints and perspectives increase the pool of ideas. Different perspectives arise from a variety of values and communication styles. Ethnic diversity can lead to greater creativity, improved problem solving and better decision making (P. Richardson, 2005). Decision making can be improved in ethnically diverse teams “by using multiple points of view, increased availability of knowledge and skills, and constructive conflict” (Shachaf, 2008, p. 115). Innovation is also enhanced through the benefits of knowledge about different markets. Winkler and Bouncken reported that many of those they interviewed in five globally innovative and culturally diverse teams commented on the benefits of different viewpoints, understanding about global markets and knowledge on how to communicate with stakeholders from different cultures (2009).

While team diversity in general is often considered to improve innovation (Jehn, Northcraft, & Neale, 1999; Post, et al., 2009; van Knippenberg & Schippers, 2007) research into the effect of ethnic diversity is mixed (Brandes, et al., 2009; J. Chen, Sun, & McQueen, 2010; Horii, et al., 2005;

P. Richardson, 2005). Cultural diversity[3] improves a team's performance in long term groups, particularly for complex tasks which require different solutions to be generated (Tadmor, et al., 2012; Watson, et al., 1993; Winkler & Bouncken, 2009). However, cultural diversity in newly formed groups tends to inhibit team performance as it takes time for trust relationships to form (Watson, et al., 1993, p. 374). In a study of 412 individuals working in 87 different work groups with varying degrees of ethnic diversity, it was found that learning could take place more quickly in ethnically diverse groups (Brodbeck, Guillaume, & Lee, 2011). However the benefits of different learning styles and methods associated with the variety of cultural norms and values were to some degree counteracted by ethnic diversity hindering learning as team members were less likely to ask questions, or ask each other for help when they were in a group of people with dissimilar ethnicities. The results showed there was increased individual learning in ethnically diverse teams, but only where the team was not predominantly Anglo-Saxon (Brodbeck, et al., 2011).

Conflict

Conflict can occur in ethnically diverse teams due to differences in team members’ value systems which in turn can be associated with their ethnic background. In one example, conflict occurred in a team of Indians and Jamaicans in an insurance company due to a “difference in cultural views about teamwork, power relations and time deadlines" (Walsham, 2002, p. 374). The Indian development team had a work culture of high productivity and strict deadlines whereas the Jamaican members of the application team had a greater focus on interacting with users and accepted a backlog of requests for software changes. Where there are significant differences in cultural values such as

[3] Cultural diversity is being used here as a synonym for ethnic diversity because as discussed earlier culture is an important component of ethnicity, although ethnicity also includes race.

collectivism versus individualism, teams are less able to self-manage and resolve their own conflict (Kirkman & Shapiro, 2001) and the overall team performance can be impaired (Liang, et al., 2007).

The nature of the conflict arising within ethnically diverse teams is often categorised as either task conflict or relationship conflict (Horwitz, 2005; Jehn & Bezrukova, 2004; van Knippenberg, 2007). When team members disagree about how a task should be done (task conflict) but work to provide an optimal solution, this can help the team perform (Paul, Samarah, Seetharaman, & Mykytyn Jr, 2004). Relationship conflict is generally more negative and tends to arise in an ethnically diverse team with differing values (Y. Liu, Luo, & Wei, 2008). Such conflict can inhibit knowledge transfer where one ethnic group has collectivist cultural values and the other individualistic (J. Chen, et al., 2010). In this way, the effect of ethnic diversity within teams is varied and nuanced, requiring information about the broader context in order to understand the likely outcomes (Mannix & Neale, 2005; van Knippenberg & Schippers, 2007).

Communication

There are three underlying causes of communication issues within ethnically diverse teams. One is a natural inclination to communicate more freely with people in the same group as themselves – such as being in the same ethnic group (Samovar, Porter, McDaniel, & Roy, 2012). People are also more likely to listen to, and be persuaded by messages from people they consider to be more like themselves, which can include people of the same or similar ethnicity (Choi & Rainey, 2010; Mackie, Worth, & Asuncion, 1990). Increasing collectivism and trust within the team is one way to improve communication as once other team members are seen as being within the same group, they are more likely to communicate freely (Ochieng & Price, 2010). Language barriers can also occur in ethnically diverse teams as ethnicity is often associated with one's language (Brandes, et al., 2009;

Lazear, 1999; Nam, et al., 2009). In an English speaking organisation, it is generally considered to be important for all team members to have good grasp of English and be able to speak it fluently (M. Chen, 2013).

Differences in values associated with different ethnic groups can also create communication challenges (Fang & Faure, 2011; Knutson, Komolsevin, Chatiketu, & Smith, 2003). Cultural values such as power-distance can affect how team members chose to communicate. For instance, people from high power-distance cultures where there is greater respect for authority, are more likely to favour face-to-face communication with superiors as this shows greater respect than a medium such as email (R. M. Richardson & Smith, 2007). Furthermore, the medium used to communicate can be seen to have symbolic meaning and this can differ between cultures. For example, it has been posited that a Japanese person may perceive email as being used for informal messages whereas an American would be see it as more formal (R. M. Richardson & Smith, 2007). This has implications for the practice of software development because, as previously discussed, communication is a critical enabler for the performance of software development teams.

In summary, the influence of ethnic diversity is subtle but can have a major effect on the performance of teams. Task related aspects can be enhanced by ethnic diversity while relationship related aspects can be negatively affected. Software teams are increasingly ethnically diverse due to the growth in distributed software development and a globally mobile labour force. In order to examine how ethnic diversity influences the productivity of software teams, it is necessary to understand how software development productivity is measured and improved.

Measuring and Improving Software Development Productivity

Previous Attempts to Improve Software Development Productivity

Software organisations have tried many approaches to improving software development productivity with varied success. One approach has been to outsource software development to offshore organisations that claim to be able to achieve greater efficiencies based on scale and specialisation, which combined with lower wages in countries such as Russia and India, lead to lower software production costs (Boden, et al., 2009; Ehrlich & Chang, 2006; Herbsleb & Mockus, 2003; Herbsleb & Moitra, 2001). Many organisations that have moved software development offshore have been able to benefit from faster, cheaper and better software development (Šmite, et al., 2010), but research also shows that these benefits are not always achieved (Carmel & Abbott, 2007). Obstacles to achieving these benefits include “problems of cultural diversity, inadequate communication, knowledge management and time differences” (Damian & Zowghi, 2003, p. 1).

The use of repeatable, measurable and optimisable processes has also been the focus of a number of efforts to improve software development productivity. Many software engineering research projects attempt to establish process models to make software development a more predictable and productive task (Bonacin, et al., 2009). This move to process-focused software development environments is the convergence of work by a number of organisations internationally to both reduce software development costs and improve quality (Niinimaki, Piri, Hynninen, & Lassenius, 2009). Over the history of software development, a great number of software process models have been defined, each one seeking to improve software development outcomes. These software process models include the waterfall model, prototyping, evolutionary development, incremental development, spiral model and others (Sommerville, 2004).

Two significant developments in the evolution of software development processes are Capability Maturity Models and agile development. Watts Humphrey and then Bill Curtis led work at the Software Engineer Institute at Carnegie Mellon University to develop the Capability Maturity Model (CMM), and its successor – the Capability Maturity Model Integration (CMMI) with the goal of improving software development outcomes. CMM and its successor CMMI have five levels of process maturity ranging from 1 (ad-hoc) through to 5 (optimising) (Carnegie Mellon Software Engineering Institute, 2006). Many researchers and industry observers state that the higher the level of process maturity, the more productive a software development project will be (Boehm, et al., 2000b; Gibson, et al., 2006; McGuire, 1996; Putnam & Myers, 2003; Yu, et al., 2009). However, proponents of agile methods generally claim that CMMI based methods are not productive whereas effective use of agile methods results in improved productivity in some, if not all situations (Beck, 2000; Cockburn, 2002; Fowler, 2005; Highsmith, 2002; Martin, 2003). Agile proponents generally contend that the application of engineering methods, such as those based on CMMI, require software development teams to try to plan out a large part of the software process in great detail for a long span of time. This works well until things change. As a consequence, software development teams applying such methods tend to resist change, unlike those who adopt agile methods who are more inclined to embrace change. However, regardless of whether an organisation uses an agile or a plan-driven method, skilled people and a cohesive team are a significant determinant of whether software development will be productive (Alleman, et al., 2004; Boehm & Turner, 2005).

Measuring Software Development Productivity

An important aspect of improving software development productivity is the ability to measure it. A number of software development productivity metrics exist, each with different definitions of the inputs and outputs of software development. The Putnam and Myers (2003) software development productivity index uses source lines of code (referred to from here on as “lines of code”). This index uses their SLIM (Software Lifecycle Model) model of software development productivity to compare the ratio of inputs to outputs between organisations. However other researchers criticise the use of lines of code (Ghezzi, Jazayeri, & Mandrioli, 2004). Some of the shortcomings of using lines of code as a measure of software include problems with dead or inefficient code, differences between different programming languages and challenges with some modern development environments where no lines of code are produced (Ghezzi, et al., 2004; C. Jones, 2000). For example, often no lines of code are created in developing artificial intelligence software (Nemecek & Bemley, 1993). Xia, Ho and Capretz (2008) note that lines of code as a measure “is a natural artefact that measures software physical size, but it is ... difficult to have the same definition across different programming languages” (p. 3). This is because each development language requires a different number of lines of code to deliver an equivalent amount of end user functionality. Earlier development languages (for example, first generation languages) generally required more lines of code than later development languages (for example, fourth or fifth generation) to deliver the same amount of functionality (C. Jones, 1995).

An alternative to lines of code is function points, created by Allan Albrecht at IBM in the 1970s as a metric that could measure the value of software in terms of the functionality it delivers (Albrecht, 1979). The function point measure is useful as a software size metric as it can be obtained early in the software development lifecycle. It measures the functional size of the software and it is programming language independent (International Function Point Users Group, 2005). Function points are calculated by first quantifying the information processing functionality associated with a software system. This first step determines the quantity of unadjusted function points (UFP). The second step is to assess the degree of influence of fourteen Application Characteristics (AC) and rate each of them on a scale from 0 to 0.05. These ratings are added together and then added to a base constant of 0.65 to produce an adjustment factor in the range from 0.65 to 1.35. This is multiplied by the UFP count to give the adjusted function point count. Due to the benefits of this measure, Capers Jones stated in 2000 that "function points are now the preferred choice for software economic studies involving multiple programming languages and full life cycle costs” (C. Jones, 2000, p. 74). Function points are still the preferred measure of software development productivity due to the benefits outlined above, and because there is a significant body of software development productivity data that exist, the use of function points is a way of measuring and comparing productivity (P. Hill, 2010).

As the purpose of Function Point Analysis (FPA) is to provide a common sizing metric for software and not directly to estimate the cost, the fourteen ACs are not generally calibrated to changes in software development methods across the industry or to a specific organisation. Two reasons why this is a limitation of FPA is that the ACs are limited to the 14 identified in the model, and the maximum effect any of the 14 ACs can have on software size is 5%. As an example, one of the ACs is ‘reuse’. A 5% limit on the effect of reuse is inconsistent with software development projects observed in the real world (Boehm, et al., 2000b). Therefore, in most studies of software development productivity, the 14 application characteristics are omitted and unadjusted function points (UFP) are used (Boehm, et al., 2000a; P. Hill, 2010; C. Jones, 2008; Kitchenham, 1997).

The most widely used software development productivity measure is the project delivery rate (PDR). The PDR is a ratio of inputs to outputs where inputs are measured in hours of effort and outputs are measure in unadjusted function points (P. Hill, 2010). The PDR is therefore the hours it takes to deliver each function point, with a lower figure indicating a more productive project. This allows the productivity of each project to be compared with other software projects. It also allows projects to be compared with previous research into software development productivity (Lokan & Mendes, 2006) as well as industry benchmarks (International Software Benchmarking Standards Group, 2009). PDR has been widely accepted as a cross technology measure of software development productivity (Afsharian, Giacomobono, & Inverardi, 2008; Ebert & Dumke, 2007; Kakola, 2008).

When considering software development productivity, it is important to also consider product quality. Improved productivity should not be at the expense of product quality. Product in the context of software development "usually refers to what is delivered to the customer" (Ghezzi, et al., 2004, p. 3), that is, the software itself. The qualities of software are the things the user wants in the software. A narrower interpretation of software quality is the degree of defects in software (Devnani-Chulani, 1999). Given the intangible and malleable nature of software it is difficult to achieve and assess product quality unless a suitable production process is in place (Cugola & Ghezzi, 1998).

To analyse software development productivity, product quality can be considered in at least two ways. One approach is that the output of the production process is measured not just in the quantity of the output, but also the quality of the output. The alternative and more practical approach when focussing on productivity is to assume that a certain level of product quality is required and this is a factor that affects productivity (Boehm, et al., 2000b). In this way the required level of quality is a factor that influences the productivity that can be achieved. The higher the level of quality required in the software produced, the less productive the software development process will be.

Models for Analysing Software Development Productivity

Quantitative analysis of software productivity has traditionally been used to evaluate the factors that influence productivity. Cost estimation models for software development encapsulate the factors that affect software development productivity and attempt to estimate the effort required to develop software. Therefore, a software development cost estimation model can be used to analyse productivity. A number of approaches to predicting software costs exist (Boehm, 1981; Devnani-Chulani, 1999; C. Jones, 2000). Effort estimation is one such approach and is concerned with estimating the inputs (effort) required to produce a specified amount of outputs (software). One method for estimating the effort required to build a software solution is an algorithmic model where the variables considered to be the main influencers of cost are used to calculate the estimated cost (Boehm, 1981).

Algorithmic models are generally developed based on a sample of historical software projects. While the goal of an algorithmic cost estimation model is to be able to predict software project costs across multiple organisations, the accuracy of such models are generally improved by calibrating for a single organisation based on an organisation's previous projects (Kitchenham, Mendes, & Travassos, 2007; Lokan & Mendes, 2006; Mendes & Lokan, 2008). The relationship between the productivity factors and productivity are represented in some models as simple linear relationships (Albrecht, 1979; Halstead, 1977; Karner, 1993) while in others, exponential relationships are used (Boehm, et al., 2000b; Putnam, 2001). Including exponential relationships allows for the effect of some productivity factors to increase or decrease as other factors (such as project size) change. Work has also been undertaken to develop non-algorithmic models using neural networks (Huang, Capretz, Ren, & Ho, 2003; Idri, Mbarki, & Abran, 2004; Shukla, 2000) and fuzzy logic (López-Martín, Yáñez-Márquez, & Gutiérrez-Tornés, 2008; Mittal, Parkash, & Mittal, 2010;

Papatheocharous & Andreou, 2009). While these models have been found to be more accurate in some cases than algorithmic models (Ahmed, Saliu, & AlGhamdi, 2005; Muzaffar & Ahmed, 2010; Saliu, Ahmed, & AlGhamdi, 2004), they still require the productivity factors to be previously defined either through an existing algorithmic model or other research into the major productivity factors. Table 4 provides a summary of a selection of major algorithmic models of software development. None of these models include ethnic diversity as a productivity factor.

Table 4. Algorithmic software development cost estimation models

Model	Summary	Open or proprietary	Source
CHECKPOINT	Estimates hours of effort as well as risk and hardware needs	Proprietary	(Rubin, 1983)
COBRA	Combines cost benchmarking and productivity data to estimate project costs and risks	Open	(Briand, El Emam, & Bomarius, 1998)
COCOMO II	A comprehensive parametric cost estimation model using a wide range of productivity factors.	Open	(Boehm, et al., 2000b)
COSMIC	A new version of function points intended to be more relevant and future proof	Open	(Abran, Desharnais, Oligny, St-Pierre, & Symons, 2009)
Ellis	Defined productivity factors for component based development	Open	(Ellis, 1995)
ESTIMACS	Asks 25 questions about the complexity of the software and user organisation to determine hours of effort	Proprietary	(Rubin, 1983)
FULSOME	A New Zealand developed model which applies fuzzy logic to data available at each stage of a software project	Open	(MacDonell, Gray, & Calvert, 1999)
IFPUG	The most widely used form of function point analysis	Open	(Albrecht, 1979)
Halstead	Uses size, difficulty and programming effort to determine overall hours of effort required	Open	(Halstead, 1977)
PRICE-S	Has sub-models for acquisition and development	Proprietary	(Minkiewicz, 2008)
SAIC	Estimation model for component based development that focuses on end users costs of integrating new software	Open	(Karpowich, Sanders, & Verge, 1993)
SEER-SEM	Estimates hours of effort as well as other outputs such as risk and hardware requirements	Proprietary	(Galorath, 2012)
SLIM	Uses lines of code and a productivity index to calculate total hours of effort	Proprietary	(Putnam, 2001)
Use Case Points	An updated version of function points based on requirements captured as use cases	Open	(Karner, 1993)

There is some overlap between the models listed in Table 4. For example, function points derived from function point analysis are used as an input into other models such as Ellis, COBRA and COCOMO II. Some of the models presented in Table 4 **Error! Reference source not found.** are primarily functional size models (Function point analysis and COSMIC). However, these have been included as they are typically used to estimate effort by determining the size and complexity of the software and then using the number of hours per function point as a parameter to determine total hours of effort required (Boehm, 2006; P. Hill, 2010; International Function Point Users Group, 2010; C. Jones, 2008). Of the algorithmic models listed in Table 4, COCOMO II is by far the most cited with almost 1000 articles on the IEEE article database referring to this model. It also forms the basis of many other software cost estimation models. It is the most comprehensive in terms of the number of productivity factors it includes and the details it provides about the relationship of each factor to productivity.

Constructive Cost Model (COCOMO)

The Constructive Cost Model (COCOMO) estimates the units of effort required to develop software of a specified size in a specified environment in terms of function points or source lines of code. The factors that affect productivity (referred to as cost drivers in COCOMO) are then applied to the size of the software in order to estimate the total amount of effort required in hours or days. COCOMO (Boehm, et al., 2000b) represents the software development process as a socio-technical system, by explicitly representing the causal relationships between the characteristics of a software development project and the resulting productivity. The original COCOMO cost estimation model was released in 1981 (Boehm, 1981) and the revised improved version COCOMO II released in 2000 (Boehm, et al., 2000b).

The model development process adopted by Boehm et al (2000b), involved first identifying the software development parameters on which to gather data. These parameters were based on those factors which were most likely to affect software development productivity and were based on a review of the relevant literature along with a series of workshops with industry experts. Once the decision was made on which factors were most likely to affect productivity, data were gathered from 161 software projects that were completed between 1994 and 2000. The results from this data analysis were combined with independent assessments by industry experts to determine how much each factor affected productivity. Each of the key influences of productivity is rated using an ordinal scale similar to a Likert scale (Likert, 1932). For example, team cohesion is one factor in COCOMO II which accounts for the effect on software development productivity arising from challenges in synchronising the project's stakeholders, including the team members (Boehm, et al., 2000b). This can be rated from very low (very difficult interactions) through to extra high (seamless interactions). A full list of the COCOMO II variables used in this study is included in *Appendix F – List of Variables*.

Using Bayesian statistical analysis the data from the 161 software projects was combined with industry experts' assessments to provide a calibrated version of the model (Boehm, et al., 2000b; Chulani, Boehm, & Steece, 1999). Based on *a priori* data from the panel of experts and the project data gathered, the impact of each factor (such as team cohesion) on software development productivity had been determined. This resulted in a calibrated version of the model known as COCOMO II.2000 as it was released in the year 2000 and to distinguish it from the generic, non-calibrated model referred to as COCOMO II. By adopting this model development process, both the selection of the factors most likely to affect productivity, and the determination of how much each factor affects productivity have been identified through a combination of expert opinion and quantitative analysis of software projects. The calibrated COCOMO II.2000 model was tested with completed software projects that were outside of the 161 software projects used to calibrate the model. This test concluded that COCOMO.II estimates within 30 percent of the actual results 69 percent of the time (Devnani-Chulani, 1999) which is relatively accurate for a software development effort estimation model (P. Hill, 2010; Jorgensen & Grimstad, 2012; Mishra, Hazra, & Mall, 2011).

A search of relevant databases for articles that reference COCOMO returned the results shown in Table 5.

Table 5. Articles that refer to COCOMO

Database	Articles that refer to COCOMO	Number with Barry Boehm as an author	Number since 2000 (when COCOMO II was released)	Number since 2005
IEEE	996	48	657	429
Springer	378	25	351	307
ACM	285	22	246	187

Of the articles that refer to COCOMO published since 1998 (the year COCOMO II beta was released), only 159 were available to the author of this study. These 159 articles have been analysed to categorise why they refer to COCOMO. Table 6**Error! Reference source not found.** shows the results of that categorisation.

Table 6. Categorisation of why articles refer to COCOMO

Reason for referring to COCOMO	Number of articles	Explanation
Using COCOMO	52	Articles that use the COCOMO model in some way to support research.
Enhancing COCOMO	41	Enhancements to the COCOMO model such as creating new models based on COCOMO.
Assessing COCOMO	30	Explanations and reviews of COCOMO.
Using COCOMO project data	14	Data from approximately 81 projects used to create the original COCOMO model in 1981 are available publicly and therefore some researchers use this dataset for analysing their own models and research.
Calibrating COCOMO	11	Analysis of empirical data to update the ratings for the existing COCOMO productivity drivers.
Alternatives to COCOMO	6	COCOMO is referred to in the context of discussing an alternative to COCOMO.
Tailoring COCOMO	5	Includes adding or removing productivity factors (also called cost drivers) to or from COCOMO.
Total	**159**	

Deeper analysis of these articles shows no substantial criticism of the model as a whole. Some articles propose refinements or minor additions, but the use of COCOMO II as a model of software development productivity is generally accepted. As shown in Table 5, the most common reason for an article to refer to COCOMO II is to use it in some way to support other research. In one study Haaland, Stamelos, Ghosh and Glott (2009) used COCOMO II as a basis for estimating the substitution cost of free and open source software by estimating what the cost would be if it was developed commercially. In another study, Harbich and Alisch (2007) modelled the accuracy of estimation using discrete parameters and use COCOMO II as an example to demonstrate their findings. One criticism that has recently been identified is that team cohesion is not accurately represented in the model (Snowdeal-Carden, 2013). This finding is based on earlier analysis of team cohesion and investigates the use of other team cohesion measures instead of that used in COCOMO II. These alternative measures considered are the Emotional Competence Inventory (Bar-On & Parker, 2000; Richard E Boyatzis & Sala, 2004; Sala, 2002) and the Group Environment Questionnaire (Mach, et al., 2010; Rosh, Offermann, & Van Diest, 2011) which both provide a more comprehensive view of team cohesion. Given the importance of team cohesion in software development this study explores the influence of ethnic diversity on team cohesion and how that affects software development productivity, which appears not to have been done before.

Summary

This review has examined literature that indicates the ubiquity of software across all aspects of society in the developed world and the dependence on creating new software for modern advancement. However, improvements in software development productivity have not kept pace with advances in hardware or with society’s expectations. The effect of this sub-optimal productivity is compounded by the fact that roles involved in software development command a high salary or hourly rate. This high cost of software development provides barriers to implementing technological advances that benefit today’s society. As a result there has been a focus on reducing the cost of developing software and improving efficiency. Regardless of the software development method or tools utilised, a well formed and effective team is required for efficient software development and therefore team composition is a determinant of team performance.

The effect of diversity in teams has been the subject of many studies over the last 20 years and there is strong evidence showing that ethnic diversity affects how a team operates. A number of studies have found that task related aspects are enhanced by ethnic diversity. On the other hand, relationship aspects can have a negative effect in ethnically diverse teams because, based on a social categorisation perspective, people see others who are similar to them as being part of their own group, while those who are dissimilar as part of a different group. For many reasons, modern software development involves individuals from diverse backgrounds. Software teams increasingly have ethnic diversity due to more collaborative software development and the global outsourcing of software development. Furthermore, ethnic diversity is increasingly common in software teams as they are often distributed, meaning members of the team are working in different geographical locations. A number of studies have been undertaken to identify how to improve the performance of software development teams and the effect of diversity on such teams. Some studies present frameworks for analysing diversity but do not report on any actual research undertaken. In those studies that report on empirical research into ethnic or cultural diversity in software development teams, none focus on the effect of such diversity on software development productivity.

This chapter has brought together formerly disparate strands of work. Ethnic diversity and software development productivity have not previously been analysed together, leaving a research gap to be addressed. Combining these two concepts has important implications for practice as new and innovative approaches to improving productivity are significant objectives for many software producing organisations. The next chapter describes the methods used to conduct this study. The theoretical framework is outlined, including the relevant theories, the conceptual model and the justification for the methods selected. An explanation is provided of the mixed methods approach used to bring together the results to provide answers to the overall research questions.

CHAPTER 3 – METHOD

Introduction

This chapter describes the research design and methods used in this study. The research questions are first restated. The research paradigm is described and the selection of the mixed methods design is explained. This is followed by the theoretical framework which includes the relevant theories and the conceptual model used. The instruments selected for this study are described before outlining the approach used to select the sample, the sample selected and the data collection process. An explanation is provided of how validity, legitimacy and transferability have been addressed in this study and the ethical considerations are stated. Discussion of pretesting that was performed is then presented. Finally an explanation is provided of the data analysis undertaken to bring together the results to provide answers to the overall research questions.

Research Questions

This study seeks to understand the influence of ethnic diversity on the productivity of software teams. A review of the relevant literature has identified that a key factor affecting productivity is team composition and in particular ethnic diversity. However previous studies generally report that there is no direct relationship between ethnic diversity and team performance and instead ethnic diversity can influence performance through various team processes and other intervening variables. For example, the Diversity Research Network have described how diversity impacts team performance through team processes such as communication, creativity and information sharing (Kochan, et al., 2003). Other researchers have reported that task related aspects are enhanced by team diversity while relationship related aspects are negatively impacted (Mannix & Neale, 2005; van Knippenberg, De Dreu, & Homan, 2004). For this reason it is necessary to develop a nuanced understanding of the nature of team diversity (Mannix & Neale, 2005), and in particular an understanding of the role of intervening factors through which ethnic diversity influences productivity. Therefore, in order to understand the influence of ethnic diversity in software teams, it is first necessary to understand the factors that influence productivity and then examine how ethnic diversity influences those productivity factors. This gives rise to the first two research questions:

1. What factors influence productivity in ethnically diverse software teams?
2. Does ethnic diversity influence the productivity of software teams (through the factors that influence productivity)?

The influence of ethnic diversity on team performance has been found to be affected by mediating factors such as organisational context which leads to the third research question:

3. What mediating factors alter the influence of ethnic diversity on the productivity of software teams?

Research Paradigm

In order to determine the research paradigm most appropriate to address the research goals of this study, previous studies into diversity in teams in the area of IS have been reviewed. Selecting an appropriate paradigm to guide research design and execution is important as this makes explicit the assumptions adopted (Greene & Caracelli, 1997) and the basic beliefs, or axioms applied (Borland, 1990). Research within the IS domain has traditionally been dominated by the axioms of positivism or closely related paradigms (Ramiller & Pentland, 2009; Trauth, 2001; Walsham, 1995). The positivist paradigm assumes that all aspects of the universe can be objectively quantified (H. K. Klein & Myers, 1999) and that inferences from a representative sample can be generalised to a wider population (Orlikowski & Baroudi, 1991). The focus of a positivist approach is often on relationships of cause and effect between variables which are represented using numerical values (Hirschheim & Klein, 1989). Post positivism is a more recent derivative of positivism which recognises that often outcomes cannot be perfectly predicted (Lincoln, Lynham, & Guba, 2011) and that multiple versions of constructed reality are legitimate (Borland, 1990). The positivist paradigm is typically associated with quantitative methods although this is not always the case (H. K. Klein & Myers, 1999, 2001).

In the last twenty years the interpretive paradigm has gained acceptance within the IS research community (A. S. Lee & Hubona, 2009; Venkatesh, et al., 2013). The interpretive paradigm presupposes that individuals construct meaning through their interpretation of the world around them (H. K. Klein & Myers, 2001). Knowledge is formed by understanding the meaning that people assign to situations and events they encounter (Walsham, 1995).

The interpretive paradigm is considered most appropriate for this study, given the nature of the research objectives, questions and context (Venkatesh, et al., 2013). The complex and nuanced nature of human interactions and perceptions can best be understood using assumptions underpinning the interpretive paradigm (H. K. Klein & Myers, 1999; A. S. Lee & Hubona, 2009; Walsham, 1995) and this paradigm has been used in previous studies of team diversity (Chen, et al., 2010; Shachaf, 2008; Winkler & Bouncken, 2009). Venkatesh et al. (2013) also comment that "IS phenomena are socially constructed and not fully deterministic" (p. 28) suggesting that an interpretive paradigm is appropriate given that it is most suited to understanding socially constructed realities (H. K. Klein & Myers, 2001). This study also considers outcomes in terms of the productivity achieved which implies a need for measurement and variables (Black, et al., 2009).

The research questions for this study focus on the *influence* of ethnic diversity rather than seeking to prove a cause and effect relationship and such a focus indicates an interpretive paradigm (Guba & Lincoln, 1989; J. A. Maxwell, 2004; Shavelson & Towne, 2003). However, there is also a quantitative component to the research in particular the examination of productivity. Therefore, the dominant approach is qualitative and interpretive and this is augmented with the use of some quantitative data.

Mixed Methods Design

Mixed method research generally involves combining qualitative and quantitative methods (Creswell & Clark, 2011). Both qualitative and quantitative methods are being used in this study to provide richer and broader results. Caracelli and Greene (1997) argue that the use of mixed methods is a legitimate and effective means of generating “more relevant, useful and discerning inferences” (p. 19) from research. Each method provides a different perspective on the answers to the research questions and at the heart of these differences are assumptions or axioms regarding how inference is justified and how causality is defined (Harrits, 2011; A. S. Lee & Hubona, 2009; Tashakkori & Teddlie, 2010). Inference by statistical generalisation with numerical values leads to different conclusions to those gained with "analytical generalisation" (Yin, 2009, p. 43) which involves generalising the findings to a broader theory. Quantitative based causality implies predictor variables which appear to precede an effect whereas a wider examination of causality leads to questions about whether outcomes can be predicted especially in complex socio-technical systems such as software development teams (Bollen & Davis, 2009; Harrits, 2011; Kundi, 2006; J. A. Maxwell, 2004).

Elements of two major mixed method designs have been combined for this study. The first is the “concurrent embedded design” (Creswell, 2009, p. 214) with both types of the data being collected concurrently and the qualitative component being the dominant method. This type of mixed methods design has been used effectively in IS research to compare positivist and interpretive research conducted in parallel (Trauth & Jessup, 2000). The “convergence model” (Creswell &

Clark, 2007, p. 64), also known as the "convergence parallel design" (Creswell & Clark, 2011, p. 69) has also been used to provide a more complete and comprehensive understanding of the influence of ethnic diversity within teams. This model involves comparing the results from two different sources of data in terms of their convergence and has been used in previous studies of organisational diversity (Frink et al., 2003). The overarching philosophy in adopting mixed methods in this study is that of pragmatism as this supports a focus on solving real world problems and informing practice (Creswell & Clark, 2011; Tashakkori & Teddlie, 2003).

The mixed methods design has been used to gather and analyse data about historical software development projects at software producing organisations. Using only quantitative or qualitative methods would not have provided the multifaceted and multidimensional perspectives that are necessary to study the nuances of team dynamics alongside numerically based measures of software development productivity. Mixed methods have been found to be useful in a number of areas of research (Petter & Gallivan, 2004) including the field of information systems (IS) research (Mingers, 2001). For example, Ormerod (1995) used mixed methods to gain multiple perspectives on the process of IS strategy development and more recently Williams (2009) used mixed methods to investigate the relationship between internet access and social cohesion.

The use of mixed methods answers calls for researchers to apply a mixed methods approach to present a more complete understanding of the research problems (Mingers, 2001; Petter & Gallivan, 2004). In doing so, this study adds to the low percentage of IS studies using mixed methods to date which is less than 5% according to a recent review of empirical IS studies (Venkatesh, et al., 2013). A mixed methods approach can provide more complete answers to research questions than a single method (Tashakkori & Teddlie, 2008) and can offset shortcomings that individual methods can have if used in isolation (Greene & Caracelli, 1997).

In this study, mixed methods are adopted as using only one data source, or using only a single method may not provide adequate answers to the research questions. Different types of methods are required to understand the complexities of ethnic diversity in software teams and examine how this influences productivity. Both of these reasons provide sound justification for adopting a mixed methods research design (Creswell & Clark, 2011; Tashakkori & Teddlie, 2010)

Combining Qualitative and Quantitative Methods

In this study qualitative and quantitative methods are combined for three key purposes, each providing greater depth or validity to the results and conclusions. These are based on the frameworks presented by Venkatesh, Brown and Bala (2013) but also take into account similar and related frameworks presented by Greene, Caracelli and Graham (1989) and Bryman (2006). The three key purposes for combining methods in this study are corroboration / confirmation, completeness and complementarity. Other reasons for using mixed methods in this study (all drawn from the framework developed by Bryman (2006)) are credibility (the use of more than one method enhances the credibility of the findings), explanation (the results from one method helps to explain the findings from the other method), enhancement (using quantitative to augment the qualitative findings), utility (where combining the findings from the two methods is more useful to practitioners and others) and illustration (where qualitative results are used to illustrate quantitative findings). While the qualitative results do in some regards expand on the quantitative findings, when mixed methods are used for expansion this generally occurs as a sequential research design (Greene, et al., 1989; Venkatesh, et al., 2013) whereas this study employs qualitative and quantitative methods concurrently. Therefore, the three key purposes selected for using mixed methods in this study are suitably comprehensive and have maximised the benefits available from combining qualitative and quantitative methods.

Corroboration / Confirmation and Triangulation

The use of corroboration / confirmation seeks to increase the credibility of the findings from both the qualitative or quantitative data (Venkatesh, et al., 2013) and increases the credibility and validity of the results. The comparison of the results from one method to see whether they converge with the results from another method is called triangulation (Bryman, 2004; Denzin, 1997; Greene, et al., 1989; Mathison, 1988). This technique has its origins in navigation where multiple readings are used to determine an exact location (H. W. Smith, 1975) and has been used in social sciences to improve the completeness and validity of research results (Bryman, 2004). Combining methods for the purpose of corroboration / confirmation seeks to enhance the validity of the findings which is discussed in the section *Validity, Legitimacy and Transferability* later in this chapter.

Completeness

Mixed methods have been used to increase the completeness of the results, providing a more comprehensive understanding of the topic under research. This involves comparing the results from two different sources of data in terms of their convergence or otherwise and has been used in this way in previous studies of organisational diversity (Frink, et al., 2003). Improving the completeness of the results has been identified by both Venkatesh et al. (2013) and Bryman (2006) as a legitimate reason for combining methods. Enhancing the completeness of findings in this way has been used in previous studies where qualitative methods were used as the primary approach for understanding the situation and then synthesised with results obtained using quantitative methods (Breitmayer, Ayres,

& Knafl, 1993; Jick, 1979). In these studies, two sets of results were analysed to understand in what ways the results converged or whether there were significant differences. More recent IS studies have also combined methods to enhance completeness by elaborating on quantitative findings using qualitative data and methods (Hackney, Jones, & Losch, 2007; Piccoli & Ives, 2003). Mixing qualitative and quantitative methods in this way has been found to “enhance confidence” (Bryman, 2004, p. 1142) in the findings and provide a richer understanding of the implications.

Complementarity

Mixed methods have also been used in this study to obtain complementary views about the influence of ethnic diversity on productivity. This study provides answers to the research questions from different perspectives and these answers have been combined to improve the overall clarity of the results. Qualitative and quantitative results have been combined in this way in previous mixed methods studies providing more holistic findings than if only a single method was used. (Creswell & Clark, 2011; Greene, et al., 1989; Venkatesh, et al., 2013). One example is an Information Systems study by Soffer and Hader (2007) in which qualitative and quantitative results were gathered simultaneously and complementary findings combined to provide a richer understanding of conceptual modelling decisions (Venkatesh, et al., 2013).

Relevant Theories

Theories relevant to this study are general systems theory and in particular two of its derivatives – complex systems theory and socio-technical systems theory. General systems theory has been applied to analysing how an organisation functions (Flood & Jackson, 1991; G. Morgan, 2006; Skyttner, 2005), as well as to the software development process (Ludwig Von, 1972; Markus, Majchrzak, & Gasser, 2002). However, given that software development is a complex system involving a number of interdependent components, complex systems theory has also been used to model and analyse workplace processes and in particular the dynamics of teams (Arrow, McGrath, & Berdahl, 2000). Complex systems research is concerned with the dynamics of systems made up of interdependent components (M. Klein, Sayama, Faratin, & Bar-Yam, 2001) and has been used to interpret and model different aspects of the software development process (Musen, 1997). The idea of complex systems is derived from complexity theory and is used across a number of domains (Arrow, et al., 2000). In general a complex system is one which has many parts and complex behaviour (Haken, 2006). Due to this, as well as the influence of environmental conditions, the outcomes from such a system cannot be predicted with absolute certainty (Fuchs, 2013).

Another derivative of general systems theory which is relevant to this study is socio-technical systems (STS) theory. STS theory is useful for analysing systems with technical and social subsystems where the technical subsystem consists of the process and technologies employed and the social subsystem comprises the knowledge, skills, attitudes, values and needs that individuals bring to the work situation (Bostrom, Gupta, & Thomas, 2009). An important aspect of STS theory is that it stresses the reciprocal relationship between people and technology which is important in the study of ethnic diversity and productivity in software development teams. Where a system contains the characteristics of both complex and socio-technical systems, it is useful to consider the system as a complex socio-technical system to ensure relevant aspects from both complex systems theory and socio-technical systems theory are brought to bear on the research problem (Pavard & Dugdale, 2006). This interdependent set of social and technical components provides an accurate description of the complex system that is the software development process (Santos & Moura, 2009). STS theory has previously been used to analyse a number of different aspects of software development such as the effect of geographical dispersion on the efficiency of software teams (Cramton & Webber, 2005). Other examples include visualising technical and social dependencies within software development to improve outcomes (de Souza, et al., 2007), and the use of a socio-technical framework to deal with queries between programmers (Ye, Yamamoto, & Nakakoji, 2007).

Conceptual Model

Synthesis of the theoretical concepts most relevant to this research led to the construction of the model shown in Figure 2. This represents software development as a complex socio-technical system that is potentially influenced by ethnic diversity. The model shows the interdependency of the social and technical components that contribute to the complexity of the software development process. Such concepts, generalisations and assumptions can be used to develop a theoretical framework for the study (Lunenburg & Irby, 2007).

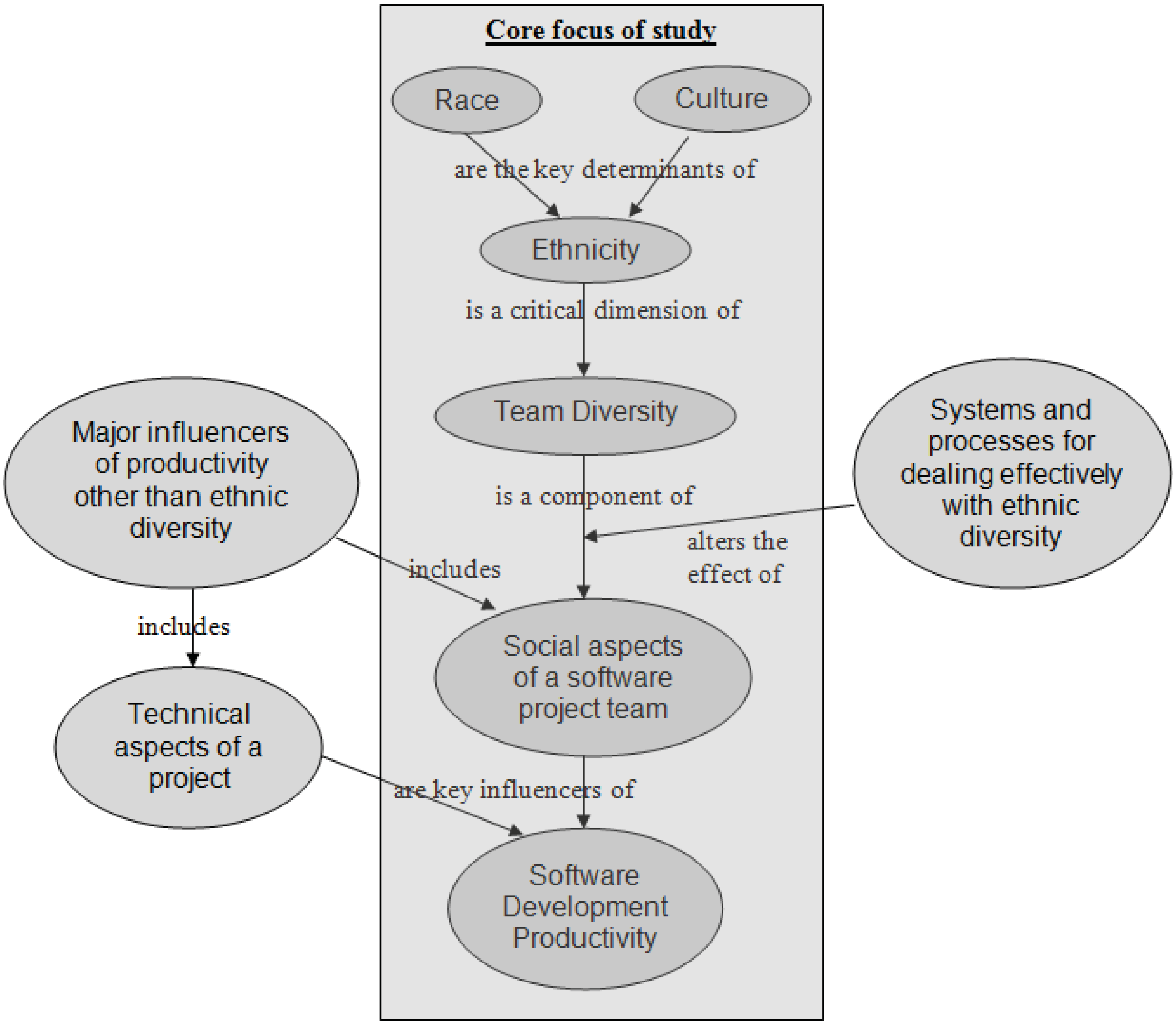


Figure 2. Conceptual model for ethnic diversity in software projects

Figure 2 models the relationship of ethnic diversity to software development productivity as a critical component of the social aspects of software development. Also modelled is the way systems and processes for managing diversity within teams significantly affects how diversity impacts team performance (Joshi & Roh, 2009; Kochan, et al., 2003; Richard, et al., 2002; Walsham, 2002). Diversity in teams has been represented through a variety of dimensions, but one of the most visible and critical types of team diversity is ethnicity (Pelled, 1996; L. Turner, 2009). As shown in the conceptual model, ethnicity is composed of both biological and socially constructed components, being a combination of race and culture (E. T. Hall, 1983; A. Smith, 1986; Winkler & Bouncken, 2009).

Selection of Instruments

To gather data on the theoretical constructs described in the conceptual model, appropriate instrumentation is required (Raykov & Marcoulides, 2010). An instrument has been defined as "a device which provides a means of extending the range and sensitivity of human sensing of nature" (Betz, 2010, p. 66). In qualitative research, it has been argued that the researcher is the instrument (Patton, 2002) and the researcher utilises certain methods to help gather and analyse data.

For the qualitative component of this study, the two methods used are interviews and an examination of project documentation. In addition to these methods that focus on projects individually, cross-project perspectives have been formed using quantitative methods. This particular approach of combining in-depth analysis of individual projects with cross-project analysis increases the credibility of results and has been used previously in mixed methods IS research (Dubé & Paré, 2003). The quantitative instruments selected are COCOMO II, project delivery rate (a software development productivity metric) and diversity indices. The instruments discussed in this section and their relationships to one another are shown in Figure 3.

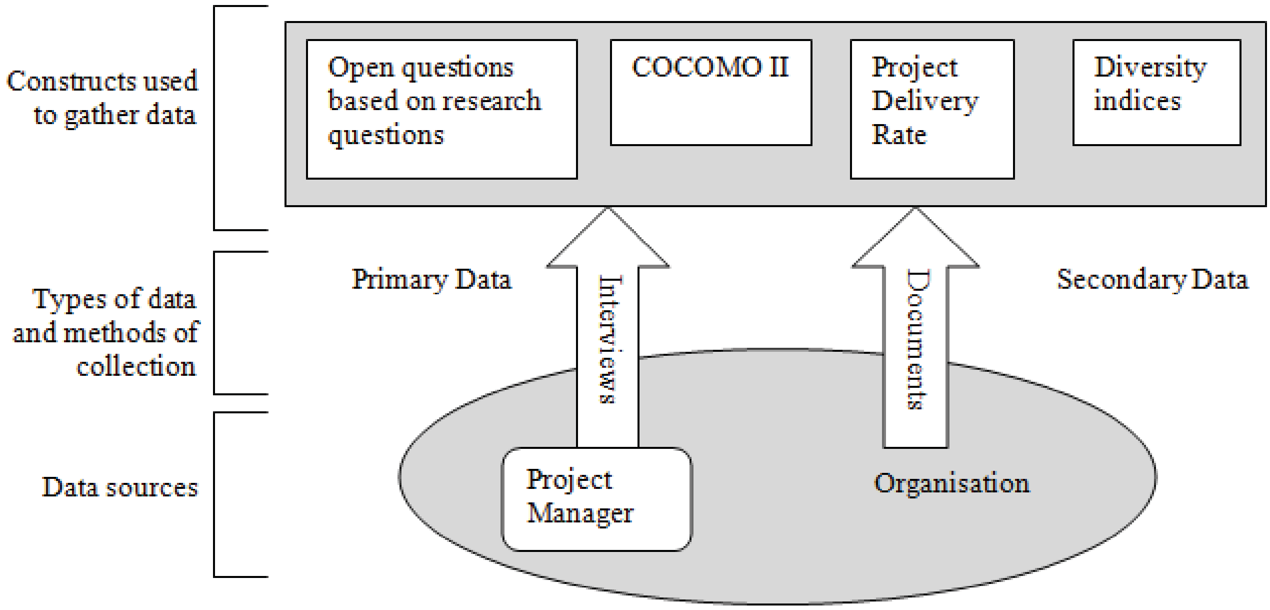


Figure 3. Instruments selected for this study

Semi-Structured Interviews

Semi-structured interviews were selected as they enable the researcher to specify the themes and broad subjects to be discussed, but allow the topics that unfold to be investigated in detail (William & Andrew, 2009). These types of interviews are the most commonly used method of gathering primary data in qualitative IS research (Benbasat, Goldstein, & Mead, 1987; Dubé & Paré, 2003) and have been used successfully to analyse software development projects (Ang & Slaughter, 2001; Kirsch & Beath, 1996). Using interviews as the primary means of gathering data provides insight into the perceptions and beliefs of key individuals about the area being researched (Wilson, 2010).

The combination of open and closed questions in the interviews enabled the interviewer to gather insight into both predetermined constructs (such as whether ethnic diversity improved problem solving on these software projects) as well as emerging themes which were not anticipated when planning the interviews (Creswell, 2009). The questions were topic and theme-centred while still recognising “that knowledge is situated and contextual” (Mason, 2004, p. 1020). This enabled relevant contextual aspects relating to the interviewee (the project manager) and their project to be brought into the frame of analysis when drawing inferences from the data. Semi-structured interviews also allow an exchange of views between interviewer and interviewee around a theme or topic (Kvale, 1996), thus enabling an active and reflexive approach to knowledge construction, rather than analysing only the initial answers given by the interviewee (Mason, 2004).

Document Analysis

Documentation is typically used to validate and expand on primary data gathered using interviews (Dubé & Paré, 2003; Eisenhardt & Graebner, 2007). Documentation was useful as it could be reviewed before embarking on an interview to provide some context. Such preparation enables the best use of the interview time by allowing some data to be gathered ahead of time and ensures the interview questions can be focussed on areas of interest (Yin, 2009).

Some of the advantages of using documentation are that documents are stable, generally precise and collecting them is relatively unobtrusive (Yin, 2003). However, there are potential risks in using secondary data which can lead to challenges to validity. These can be described as “external criticism”, concerning whether the document is genuine, and “internal criticism” (Berg, 2009, p. 271) relating to whether the document is reliable. In order to mitigate any risks arising from such criticism, document based information was cross-checked with other information sources such as interviews (Berg, 2009). Another risk of documentation is selective availability where some document was not provided or was not available. In this case, other information sources, such as follow up questions in the interviews were used to obtain any missing information to derive a more complete view of the project.

COCOMO II

To answer the research questions, it was necessary to capture and analyse factors already known to affect software development productivity in order to distinguish the influence of ethnic diversity from other factors. This allows a deeper understanding of the influence of ethnic diversity by considering whether any of the key productivity drivers are related to ethnic diversity. While ethnic diversity may not be directly related to productivity, it may be related to factors previously found to affect productivity in software teams. Therefore, existing models which encapsulate software development productivity factors have been analysed. The algorithmic software development estimation models are most relevant to this study because such models identify the variables that affect productivity and what the relationships of those variables are to the resulting productivity. From a search of suitable algorithmic models, COCOMO II (Boehm, et al., 2000b) was selected as the most appropriate base model (see the earlier section *Constructive Cost Model (COCOMO)* in *Chapter 2 – Literature Review* for details of this search). As COCOMO II uses measures which are compatible with the project delivery rate measure of productivity (described below) this enables the factors influencing software productivity to be compared to the actual productivity achieved (Boehm, et al., 2000b; Fairley, 2007; Taeho, Donoh, & Jongmoon, 2010). Furthermore, as COCOMO II includes a comprehensive list of the factors found to affect software productivity, this enables relationships between ethnic diversity and these productivity factors to be identified and examined.

The productivity factors included in COCOMO II are captured using a Likert-type scale. This is an ordinal scale as the values represent the "relative positions of the objects but not the magnitude of differences between them" (Malhotra & Birks, 2007, p. 285). In some studies, Likert scales have been treated as interval scales on the basis that the relative distance between values is approximately even (Brown, 2011) although in many other studies they are treated as ordinal scales (Allen & Seaman, 2007; Brown, 2011; Clason, Dormody, & Scales, 1994; Jamieson, 2004). In the case of COCOMO II, the assumption that the relative distance between values is approximately even cannot be justified (Boehm, et al., 2000b) and are therefore ordinal. A full list of the COCOMO II variables used in this study is included in *Appendix F – List of Variables*.

Project Delivery Rate – Productivity Measure

A measure of productivity was required to operationalise the concept of software development productivity defined in the conceptual model. A number of software development productivity metrics exist but the project delivery rate was selected as it allows the productivity of each project to be compared with other projects in the sample and has been widely accepted as a cross technology measure of software development productivity (Afsharian, et al., 2008; Ebert & Dumke, 2007; Kakola, 2008). It also allows the projects analysed to be compared with previous research into software development productivity (Lokan & Mendes, 2006) as well as industry benchmarks (International Software Benchmarking Standards Group, 2009). The project delivery rate is an interval scale as the difference between each value is constant (Malhotra & Birks, 2007).

Normalised hours of effort and productivity were used instead of the raw number of hours provided by the organisations in order to improve the comparability between projects and with industry averages. This was required as not all of the projects had effort data (that is, the number of hours spent on the project) for all six phases of the software development lifecycle (planning, specification, design, build, test, implementation) (P. Hill, 2010). Some projects had effort data for all phases, but others had effort data that covers only some of the phases. To help make them comparable, hours of effort were normalised for projects that did not include one or more of the phases to estimate the effort that would be expected if they did include all the phases.

This normalisation of hours was done by establishing the typical distribution of effort by phase. Then for projects that omitted one or more phases their effort was extrapolated under the assumption that the phases they missed out would conform to the typical profile. For example, the planning phase is on average 7.2% of the total project team effort. If a project collected the hours of effort data for most of the project phases, but did not collect and report the hours of effort spent on planning, it was assumed that its reported effort is approximately 92.8% of the total effort for the whole project. Therefore, to calculate the total effort for the project, 7.2 % is added to the hours collected to estimate the total hours spent on project. In this example, the normalised effort for the project is computed by scaling up the reported effort by 100/92.8 or 107.75% in order to compensate for the planning phase which is not included in the hours reported. This procedure involves making assumptions in order to determine the normalised effort for projects that do not provide effort data for one or more phases. While there is a small risk that the total hours of effort recorded for the project could be slightly imprecise, this risk is more than offset by the improvement in data comparability (P. Hill, 2010). Furthermore, this is the recommended practice when software project data is provided from different organisations (International Software Benchmarking Standards Group, 2007). Table 7 shows the average breakdown of effort by phase, based on analysis of the 4100 projects in the ISBSG database in 2007. The greatest effort is in the build phase at 37.8% with the least effort in the plan phase at 7.2%.

Table 7. Average breakdown of effort by phase (Jiang, Jiang, & Naudé, 2007)

Phase	**Average amount of effort spent on each development phase**
Plan	7.2%
Specify	15.9%
Design	12.9%
Build	37.8%
Test	17.6%
Implement	8.6%
Total	**100%**

Ethnic Diversity Index

Measures of ethnic diversity were required in order to operationalise this component of the conceptual model. Participants in this study were asked to nominate their own ethnicity and estimate team members' ethnicities using the five level 1 ethnic categories adopted by Statistics New Zealand (2005a). The five level 1 ethnic categories are shown in **Error! Reference source not found.** with the full list of ethnic sub-categories presented in Ethnicity New Zealand Standard Classification (Statistics New Zealand, 2005a) which is reproduced in Appendix G – List of Ethnic Groups. Where an ethnicity was identified at a more detailed level than the five level 1 ethnic categories, the full list was used to determine the level 1 ethnic group to which an individual belonged or was assigned. For example, for someone who was Fijian Indian, consulting the full list shows that according to the Statistics New Zealand list, Fijian Indian is considered to be in the level 1 category of Asian.

Table 8. Level 1 ethnic categories (Statistics New Zealand, 2005a)

No.	Ethnic category
1	European
2	Maori
3	Pacific Peoples
4	Asian
5	Middle Eastern / Latin American / African

Following previous work on measuring diversity in organisations, Blau's index of heterogeneity (1977) has been used. This index can be expressed as $1 - \sum P_i^2$, where P is the proportion of the group for each ethnicity and i is the number of ethnicities. Blau describes this index as the "operational measure of heterogeneity" (1977, p. 78) and is the probability of two people selected at random being from different groups. Recent work on diversity has identified that this index works well when there are at least as many members in the group as there are categories, but can produce inaccurate results where there are more categories than members (Biemann & Kearney, 2010). For example, the results would be inaccurate where there were five possible ethnicities but fewer than five people in the team. Therefore, a modified Blau's index is used which is unbiased towards either small or large teams (Harrison & Klein, 2007). The modified Blau's index used in this study is that proposed by Harrison and Klein (2007) and shown to be accurate by Biemann and Kearney (2010) which is "$1 - \sum(\frac{N_i(N_i-1)}{N(N-1)})$, where N_i is the absolute frequency of group members in the ith category and N is the total number of group members" (Biemann & Kearney, 2010, p. 584).

The range of values could in theory be from 0 (all one ethnicity) through to 1 (a team of five people with each person being from a different ethnic group). Richard et al. (2002) suggest that a value below 0.25 is low diversity, 0.25 to 0.5 implies moderate diversity and a value above 0.5 represents a high level of diversity. This diversity index is an interval scale because the difference between each value is constant (Malhotra & Birks, 2007). Blau's index of heterogeneity has been widely used in studies of ethnic diversity (Hambrick, et al., 1996; Richard, Barnett, Dwyer, & Chadwick, 2004). Furthermore, this type of index is the most common index used in literature on diversity and economic performance (Alesina & La Ferrara, 2005).

Validity, Legitimacy and Transferability

Validation is recognised as "a major issue in mixed method research" according to Venkatesh et al. (2013, p. 15). While the tests for quantitative studies are widely used and understood, tests for qualitative and mixed method studies are more varied (Golafshani, 2003; Onwuegbuzie & Johnson, 2006). Legitimisation rather than validity is often more salient to qualitative research and this describes the degree to which the inferences "are credible, trustworthy, dependable, transferable, and/or confirmable." (Onwuegbuzie & Johnson, 2006, p. 52).

A number of different validity or legitimisation criteria for qualitative research are proposed by different researchers (Golafshani, 2003; Healy & Perry, 2000; Kirk & Miller, 1986; Morse, Barrett, Mayan, Olson, & Spiers, 2002). For example, Golafshani (2003) discusses the value of triangulation within a qualitative study such as the use of interviews, observations and recordings to ensure research participants' intended meanings are understood. By contrast, Morse et al. (2003) focus on validity in qualitative studies in terms of the depth and rigour employed by the researcher and they state "to validate is to investigate, to check, to question and to theorise" (p. 19). Yin (2009), however, suggests that the traditional concepts of validity applied to quantitative studies are often still used for evaluating qualitative and mixed method research as these tests are most widely understood and recognised. Yet another perspective is that the concepts of validity and reliability are not necessarily relevant to qualitative studies (Caroline, 2001; Lincoln, et al., 2011; J. A. Maxwell, 1992). Regardless of the approach adopted to validity in qualitative research, validity in mixed methods is not simply the sum or the intersection of the qualitative and quantitative validity tests – rather it has been proposed that there are tests which apply only to mixed methods (Johnson & Onwuegbuzie, 2004; Onwuegbuzie & Johnson, 2006; Venkatesh, et al., 2013). Of these mixed methods specific tests for validity, those proposed by Venkatesh et al. are the most current and incorporate previous work, such as that by Tashakkori and Teddlie (2008). The Venkatesh et al. (2013) framework separates issues of validity into design quality and explanation quality. Design quality is composed of design suitability, design adequacy and analytic adequacy, while explanation quality is made up of quantitative inferences, qualitative inferences and meta-inferences. The remainder of this section uses the Venkatesh et al. framework to explain the validity, legitimacy and transferability of this mixed methods study.

Design Quality

The suitability of the research design for the research questions is the first component of the design quality aspect in the Venkatesh et al. (2012) framework. The research questions relate to the influence of ethnic diversity in software teams and therefore the nature of human interactions and perceptions. These areas of study can best be understood using qualitative methods. However, there are components to the research questions which imply a quantitative paradigm, in particular the examination of software development productivity.

The second component of design quality is design adequacy which is described as "the degree to which the design components are implemented with acceptable quality and rigour" (Venkatesh, et al., 2013, p. 24). In qualitative research, this can be achieved through measures such as consistent application of appropriate interview protocols and the use of qualitative data management tools (Yin, 2009). Therefore, one strategy employed to ensure reliability in this study has been the use of a database (NVivo) to transcribe and manage the recorded interviews and the coding of themes. Transcribed interviews were then sent to the interviewee for verification. Interviewees were asked to review and check the text of their interviews once they were transcribed to ensure their responses were accurately recorded. Appropriate protocols for interviews and data gathering were used and an "audit trail" (Patton, 2002, p. 93) maintained to verify the rigour of data collection activities.

To ensure design adequacy for the quantitative component of the study, measures have been taken to make sure the "the constructs of theoretical interest are fully operationalized in the research" (Hoyle, Harris, & Judd, 2002, p. 33). The theoretical constructs relevant to this study are identified earlier in this chapter in the section *Conceptual Model*. Data were gathered from multiple sources, that is, interviews and existing project documentation. Furthermore, an existing model of software development productivity was used (COCOMO II) to identify operational measures, combined with diversity measures based on those used in previous studies (Alesina & La Ferrara, 2005; Blau, 1977). To ensure rigour, the statistical software package (SPSS) has been used to manage and analyse statistical data.

The third component of design quality is analytical adequacy which addresses "the degree to which the data analysis procedures / strategies are appropriate and adequate to provide plausible answers to the research questions" (Venkatesh, et al., 2013, p. 24). To ensure the study had analytical adequacy for the qualitative component, a number of strategies were used. Interviews were carefully coded to identify whether there were re-occurring patterns across projects. After coding the first two interviews, the coding was reviewed with two other experienced researchers to get feedback on the coding approach adopted. When considering explanations for the results observed, rival explanations were also explored and addressed (Creswell, 2009; Yin, 2009).

To ensure analytical adequacy in the quantitative analysis, the existing model of software development productivity used (COCOMO II) has been previously tested to show that it represents causal relationships (Boehm, et al., 2000b; Devnani-Chulani, 1999). In similar studies examining factors affecting software development productivity a combination of correlation analysis, followed by multiple linear regression analysis has been used (Baresi & Morasca, 2007; Boehm, et al., 2000b; Marban, Menasalvas, & Fernandez-Balzan, 2008). As the sample for this study was not large enough to support multiple linear regression analysis, only correlation analysis was used. Therefore, care has been taken to ensure the prerequisites for correlation analysis have been met.

Explanation Quality

Explanation quality in the Venkatesh et al. framework covers the criteria for the quality and credibility of the inferences of the study. This is "the degree to which credible interpretations have been made on the basis of obtained results" (Venkatesh, et al., 2013, p. 24). In mixed methods research this includes the justification for statements made regarding whether the results from the study can be generalised to another context (Teddlie & Yu, 2007) and in qualitative research, this is typically referred to as transferability (Lincoln & Guba, 2000; D. Morgan, 2007; Tashakkori & Teddlie, 1998). A number of strategies have been used in this study to draw credible and sound conclusions about the transferability of the qualitative results. The qualitative results provide details regarding how the findings from the projects studied can be transferred to other projects or organisations. This is done in part by including the details and characteristics of each project, which is provided in *Appendix I – Data for All Variables for All Projects*. The capture and reporting of the details of each project helps to analyse the scope of transferability (Walsham, 1995) but is balanced with the requirement for participating organisations to be unidentifiable.

The explanation quality of the quantitative results has been maximised through a number of measures. The first is through explicit statements regarding the scope to which the quantitative results can be generalised. As the sample is relatively small and is unlikely to represent the entire software development industry, conclusions drawn relate to relationships that exist within the sample of projects studied. Correlation analysis was used to identify significant associations between variables in the sample as this is an effective technique and is a frequently used measure of association between variables (Cohen, 1988; Hoque, 2006). While the use of correlation provides no statistical evidence of causal relationships, various explanations are discussed and considered in the analysis of the data. In some cases these explanations are based on previous research which has demonstrated a causal relationship. In other cases, the explanations are based on other additional information available, such as that gathered in the project documentation and interviews. This reflects a departure from a purely positivist based research approach (Feuer, Towne, & Shavelson, 2002; Henson, Hull, & Williams, 2010) and instead provides a more interpretive, context-aware understanding (Harrits, 2011; J. A. Maxwell, 2004) of the inferences from the quantitative results.

The final aspect of explanation quality in the Venkatesh et al. framework covers the inference quality of the integration or synthesis of the qualitative and quantitative results. This quality criterion is composed of three components. The first is the quality of the integration of the qualitative and quantitative results. For this study, dominant themes from both the qualitative and quantitative results have been integrated in order to provide more comprehensive and robust answers to the research questions. Major findings from each method were analysed and this has helped explain or expand on a finding from the other method. This has allowed greater depth of understanding and more "theoretically consistent meta-inferences" (Venkatesh, et al., 2013, p. 24) to be developed.

The second component of the quality criterion *integrative inference* is inference transferability. A number of the measures used to enhance the inference quality of both the qualitative and quantitative results are also used for the meta-inferences arising from the synthesis. This includes “analytical generalisation” (Yin, 2009, p. 43) through conceptual analysis (A. S. Lee & Baskerville, 2003; A. S. Lee & Hubona, 2009; A. S. Lee & Nickerson, 2010). Analytical generalisation involves generalising the findings to a broader theory (Yin, 2009) and requires analysis of the results in the context of relevant theories (H. K. Klein & Myers, 1999). This type of generalisation can usually be further supported by replication logic where the findings from one study are replicated in other situations (Eisenhardt & Graebner, 2007; Yin, 2009) and this recommendation is therefore included in the section *Future Research*. The inference transferability of the integrated results is also enhanced by the capture and reporting of the details of each project as that helps to analyse the scope of transferability (Walsham, 1995). Furthermore, combining the results with the goal of corroboration / confirmation (Bryman, 2004; Greene, et al., 1989; Mathison, 1988) improves the strength and credibility of the findings and therefore the study's inference transferability.

The third and final component of the quality criterion *integrative inference* is integrative correspondence. This is "the degree to which meta-inferences from mixed methods research satisfy the initial purpose" (Venkatesh, et al., 2013, p. 24), or put another way, how well the meta-inferences address the research questions. The qualitative and quantitative components by their very nature focus on different aspects of the research questions and yield different types of findings (Morse & Niehaus, 2009). However, as both components focus on answering the research questions, albeit from different perspectives, this helps ensure the meta-inferences are also focussed on the research questions when the two sets of results are integrated.

Data Collection

This section explains the data collection activities undertaken in this study and begins with a discussion of gathering the documentation and the interview process. This is followed by a discussion of the pretesting conducted to improve the validity of the study and the quality of the results. Finally, this section provides a discussion of the ethical considerations covering the approval process and management of ethical issues throughout data collection.

Sample Selection and Description

The approach used to select a suitable sample was one of convenience or opportunity. This approach has been used before in gathering detailed data from software producers given the challenges that exist in recruiting participating organisations (Verner, Sampson, Tosic, Bakar, & Kitchenham, 2009). Convenience samples have been used in a number of studies where participants are not selected at random, but are recruited by seeking willing volunteers (Hultsch, MacDonald, Hunter, Maitland, & Dixon, 2002; Pruchno et al., 2008). While a convenience sample may result in some degree of bias, it allows the discovery of context-relevant insights into phenomena (Burns & Grove, 2005). By capturing the characteristics of the sample used, any potential bias could be identified and the context of the results discussed in detail (Fink, 2003). Samples have been selected in this way in previous qualitative and mixed method studies to present either representative or comparative findings (Teddlie & Yu, 2007) and is one of the approaches suggested in the literature on sample selection (Simons, 2009; Verner, et al., 2009; Yin, 2009).

Due to the density of data generated in interviews, sample sizes for interview-based research tend to be relatively small (Todd & Benbasat, 1987). As a result, in predominantly qualitative studies such as this one, there is a trade off between saturation of information, “where you’ve heard the range of ideas and aren’t getting any new information” (Krueger & Casey, 2009, p. 21) and achieving a representative sample (Teddlie & Yu, 2007). Hofstede et al, also comment on striking an appropriate balance in determining sample size to provide sufficient data for both qualitative and quantitative analysis. They comment that “twenty units (teams) was a small enough number to allow studying each unit in depth, qualitatively [and] at the same time, it was large enough to permit statistical analysis of comparative quantitative data across all cases.” (Hofstede, et al., 1990, p. 290). For this mixed methods study into diversity in software teams, the sample size ensures the study is both feasible and provides meaningful results.

Senior IT managers from twenty four software producing organisations from across the North Island of New Zealand were approached and invited to participate in the study. After the initial approach, a follow up email was sent to those who had not responded and further reminders were sent to each contact approximately every two weeks, up to a maximum of three follow up emails. Where possible, an attempt was also made to telephone those managers who did not respond to emails. The initial contact email and the accompanying letter are included in *Appendix A – Covering Email for Invitation to Participate* and *Appendix B – Invitation to Participate*. Of the 24 organisations approached, one declined, three initially accepted but later withdrew and 13 did not reply. Seven software producing organisations from Wellington, New Zealand agreed to participate – three government organisations and four non-government organisations. Of the four non-government organisations, two were primarily software producers. From these participating organisations, a sample of 19 software development projects was selected through discussion with the originations’ representatives based on which projects would be suitable for this study. Data were gathered on the 19 software projects using interviews with 11 project managers combined with a review of existing project and system documentation. Table 9 provides a summary of the data collected from each organisation.

Table 9. Summary of data collected

Organisation	Government	Project managers interviewed	Projects	Documents received
A	No	1	6	219
B	No	0	2	103
C	No	1	1	8
D	No	3	3	11
E	Yes	4	5	306
F	Yes	1	1	8
G	Yes	1	1	0
	Totals:	**11**	**19**	**655**

The unit of analysis is software development projects, of which there are 19. Each project had a different project team although in some cases, individuals were members of more than one project team. Of the 19 project teams studied, 14 were ethnically diverse, one, which was also the smallest team, was not ethnically diverse. In the remaining four projects, it was not possible to gather data on the ethnicity of the project team members. The data relating to the 14 teams that were ethnically diverse have been used for the qualitative analysis while the data relating to all 19 projects were included in the quantitative analysis of the key factors influencing productivity.

Gathering Documentation

Data collection began with a request to participating organisations for documentation on the projects they had nominated for analysis. The request for documentation sent to each organisation is shown in *Appendix E – Request for Documentation*. Documentation was obtained in a number of ways. The most common was for the organisation to email it to the researcher. In other cases, the researcher visited the organisation and went through the available documentation with the research participant. After discussing the available documentation, the documents that were most relevant to addressing the research questions were copied to the researcher's memory stick to enable them to be analysed over time. Some research participants provided printed documents and allowed the researcher to take those copies of the documents away for analysis.

The focus of the analysis was on the influence of ethnic diversity in software development productivity, establishing what other factors influenced productivity and identifying how the organisational context affected the outcomes. The system documentation provided information about the system that was developed or enhanced such as purpose, size and complexity. Project documentation provided further information such as total project effort, project duration and the challenges that were encountered during the project.

Obtaining the documentation served two main purposes. One was to provide the researcher, who was also the interviewer, with a broad understanding of each project prior to the interview. This enabled the interviewer to make the best use of the participants' time in the interviews as many of the basic questions (such as what the project was producing and what dates it started and finished) were already covered. The questioning could then be focused on examining the factors affecting productivity, how ethnic diversity influenced productivity and what other factors made a difference. Furthermore, specific questions arising from the documents reviewed could then be raised with the project manager in the interview which either clarified anything that was unclear or revealed information that would not have been uncovered if the interviews were conducted without first reading the project documents.

The second purpose of obtaining the documentation was to calculate the productivity achieved on each project. This required the hours of effort spent on the project to be determined. Often this information was drawn from multiple documents and required careful consideration of what aspects of the project were included in the time recorded in order to ensure the productivity rates were comparable across projects. In some cases the hours of effort were not captured, but other information could be used to derive the hours of effort. For example, in many cases the financial cost of the project was captured and by using other information obtained in the documents about the hourly rates for each team member, the total hours of effort could be calculated. To determine the productivity of the project, it was also necessary to determine the size of the software developed, or the size of the software change. This was determined by analysing the functionality changed or delivered using function point analysis. To count the function points, it was necessary to read either the requirements, design documents, or a combination of both and quantify each function according to the function point analysis rules (International Function Point Users Group, 2010).

Each document was reviewed to determine which ones were relevant in gathering the information described above. As a large number of documents were received, each document was recorded in a spreadsheet and a classification system was used to aid the analysis process. When documents were received from each organisation, an initial assessment of each document was used to classify it, determine its usefulness and make any notes for future reference. A second pass then examined the key documents in order to better understand the organisation and the software project and to answer as many questions as possible before the interview to ensure the interview time was put to best use. This provided an indication of areas to probe into more deeply at the interview. Also any information gaps or questions raised by the document analysis were noted to be brought up at the interview. The documents that provided details of the software developed and the hours of effort were analysed in order to calculate the productivity rate achieved on the project. These calculations and any underlying assumptions were documented and reviewed with the project manager at the interview in order to validate this information.

The types of documentation received for the projects in each sector (that is, government and non-government) are shown in Table 10. Documentation was provided for 18 of the 19 projects studied as the documentation for one project was not made available. For this project, sufficient information was gathered in the interview to compensate for the missing documentation.

Table 10. Types of documents received

Document Type	**Six Government Projects**	**12 Non-government Projects**
Project overview	25	87
Requirements	117	69
Design	8	32
Testing	0	19
Completion	2	22
Project status reports	101	61
Effort	36	1
Minutes	1	0
Function point count	24	0
Executables	0	50
Total	**314**	**341**

Interviews

Primary data was obtained using semi-structured interviews with software development project managers. Project managers were interviewed rather than individual team members as many team members had either left the project or no longer worked in the organisation. Project managers play a key role in determining the team composition (Sebt, et al., 2010) and have a good understanding of the factors that affect the productivity of a project (Ehsan, et al., 2010; Wang, 2009). Previous studies into software project teams have involved interviewing project managers

(Hsu, Shih, Chiang, & Liu, 2011; Reed & Knight, 2010; Taylor & Woelfer, 2009). Project managers meet the criteria for being a key informant identified by Marshall (1996) as they are experts in the area under study and able to provide insight into what has occurred. Therefore project managers were the best source of information about the performance of the projects and the influence of ethnic diversity.

To develop an appropriate set of questions to address the aims of the research, the interview was divided into four sections. The first gathered basic information about the project manager and the organisation. This included the project manager’s gender, ethnicity, age group and how long they had been managing projects. The first section also included questions about the organisation such as its size and whether software development was its primary activity. The second section included summary information about the project such as its length, purpose and the development method used. The interviews then moved to open questions derived from the research questions and relevant themes from the literature. These questions explored how productive the project manager believed the project was and what factors affected the productivity of the project. Questions were asked about the team’s diversity, including the ethnicity of each team member. The project managers were asked their opinion of the influence of ethnic diversity on the team’s performance, including productivity. Finally the fourth section moved to closed questions using the COCOMO II questionnaire (Boehm, et al., 2000b). In this way, the semi-structured interviews began with open questions (once the basic project information had been gathered) before moving to closed questions based on previously developed questions about the factors affecting software development productivity. This was to ensure that interviewees had the freedom to tell their story while still allowing the interviewer control to ensure that all interviewees were being asked the same questions (Mason, 2004). The interview guide is shown in *Appendix H – Interview Guide*. A copy of this guide was given to the interviewee at the start of the interview. Interviews were recorded, transcribed and then coded using the qualitative software package NVIVO.

Due to the political climate in some organisations, it has been found that IT project managers sometimes feel unable to admit shortcomings or failures in projects (Cerpa & Verner, 2009). Therefore, project managers may provide responses which limit or suppress discussion of negative aspects of their projects. They may also adapt their responses by telling the interviewer what they think the interviewer wants to hear (Miyazaki & Taylor, 2008), which is generally referred to as interviewee or participant reflexivity (Riach, 2009). To mitigate both these potential biases measures were taken to develop the participants' trust. They were provided with written assurance that their responses would be confidential and anonymised in the consent forms signed by participants and the researcher / interviewer. Most interviews were conducted in closed door offices although in two cases offices were not available. In these two cases, interviews were conducted in a space in the open plan office, away from other staff.

Questions were "loosely structured and open to what the interviewee feels is relevant and important to talk about" (Alvesson, 2003, p. 13). Asking questions in this way reduced the likelihood that interviewees simply told the interviewer what they thought they wanted to hear. These types of measures have been shown to help provide a "rich account of the interviewee's experiences" (Alvesson, 2003, p. 13), improving the quality of findings (Bryman, Bresnen, Beardsworth, & Keil, 1988; Sands, Bourjolly, & Roer-Strier, 2007) and overall research validity (W. A. Hall & Callery, 2001). After each interview, interviewees were provided with their interview transcript for review and they were given the option to make any amendments they wished.

Pretesting and Refinement

The questionnaire used for the interview was cognitively tested and refined. A project manager who was willing to provide feedback regarding the interview questions was selected for the pretest. After the interview, feedback was sought from the interviewee which was used to make improvements to the interview questions and process. As a result improvements were made to the questions to better align with the research aims and follow up questions were developed for each of the main questions which were used to seek further information from the interviewees. Pretesting has been used previously in IS research to improve the quality of the results (Boudreau, Gefen, & Straub, 2001). More specifically, pretesting helps to ensure the questions are meaningful to respondents (Qureshi & Rowlands, 2004) and minimise errors arising from potential ambiguity of the questions (Harris-Kojetin, Fowler Jr, Brown, Schnaier, & Sweeny, 1999).

To strengthen the accuracy of interview coding after gathering documentation and conducting interviews for the first two projects, some trial analysis and coding was completed. The interview analysis, along with associated coding of themes was reviewed to consider how well the interview approach, questions and coding would meet the goals of this study. This allowed refinements to the analysis approach proposed and clarified how the two key systems being used (NVivo and SPSS) would be integrated to analyse the data from different perspectives. For example, as there was an overlap in the data that each system could hold about the projects, a decision was made that the NVivo database would be the master. This was because NVivo could hold most of the information required about each project (both quantitative and qualitative). Furthermore it was found to be relatively straight forward to export data from NVivo and import it into SPSS.

Ethical Considerations

This research has been undertaken in accordance with the relevant code of ethical conduct and with approval from the appropriate ethics committee. The *Massey University Code Of Ethical Conduct For Research, Teaching And Evaluations Involving Human Participants* (Massey University, 2010a) requires participation to be informed and voluntary. In this study, volunteering organisations were informed about key aspects of the research to ensure they were giving informed consent to participate. Agreements were made with participants, individuals and organisations selected to ensure participants were fully informed and consenting to the research. *Appendix D – Consent Form* shows the consent form signed by both the researcher and each participant. A copy of the signed form was provided to each participant for their records. As part of ensuring informed consent, the participants were informed about the study's purpose, the benefits of the study, the right for participants to withdraw at any point, that responses were confidential, and how the data were safe guarded. These are the ethical safeguards recommended by the American Psychological Association (2002) for research involving human participants.

Results about an individual organisation have been shared only with that organisation while aggregated results will be made public through the release of this thesis and articles that are published as a result of this research. Raw data about individual organisations were encrypted and password protected, with the password held only by the researcher and his supervisors. This information was communicated to participants in the invitation to participate (shown in *Appendix B – Invitation to Participate*) and the information sheet (*Appendix C – Information Sheet*).

Based on the guidance provided by Massey University human ethics procedures (Massey University, 2010b), a low risk notification (LRN) for this study was submitted and approved for noting on the university's LRN database.

Data Analysis and Synthesis

This section discusses how the qualitative and quantitative analysis was conducted followed by an explanation of the synthesis process whereby the results from the two approaches were combined. Figure 4 summarises the data analysis and synthesis process for this study conducted in parallel using both qualitative and quantitative methods.

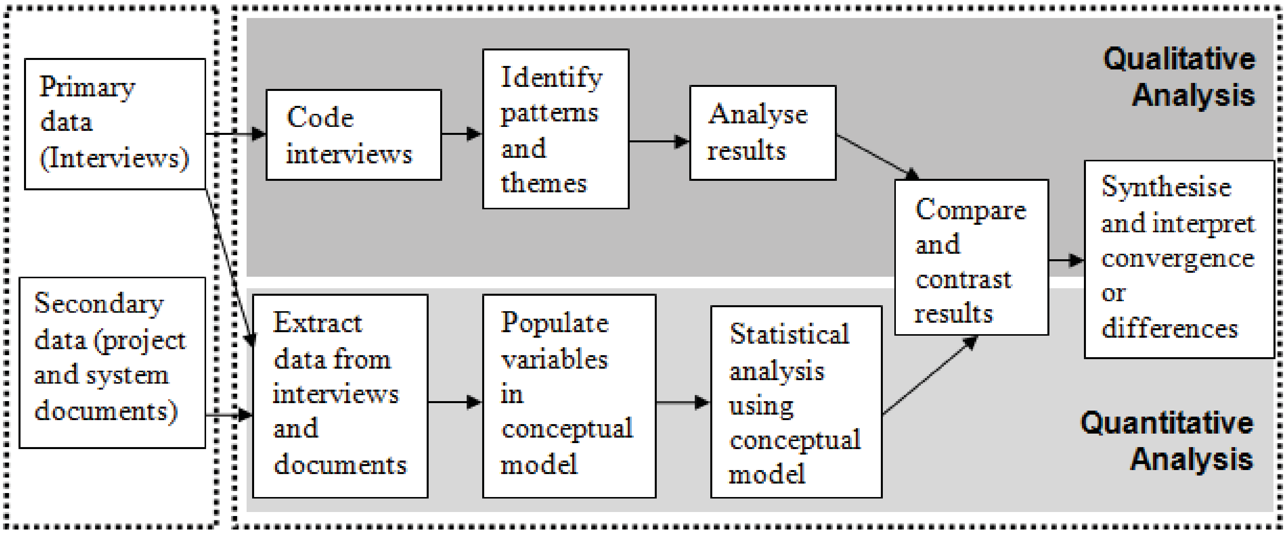


Figure 4. Data analysis and synthesis process, based on (Creswell & Clark, 2007)

Qualitative Analysis

Thematic investigation of the interviews was used first to examine the project managers' views on the influence of diversity in general and then more specifically on ethnicity. Themes were identified regarding the project managers' views of the major influences on software development productivity and the influence of ethnic diversity. Having analysed the manifest themes, latent themes were identified through interpretive analysis (R.E. Boyatzis, 1998) and were investigated as they convey “the underlying meaning of [the] narrative” (Tashakkori & Teddlie, 1998, p. 122). The latent themes identified in the interviews related to the project managers’ world view and beliefs implicit in what they talked about and the language they used. This resulted in multiple hierarchical trees of interview themes in NVivo which led to the development of concept maps (Moon, Hoffman, & Novak, 2011). An example of a hierarchical tree used is shown in *Appendix J – Coding Frequency of Productivity Factors* and a concept map is shown in *Appendix M – Concept Map of Interview Themes*.

The general approach to interview analysis was to focus initially on meaning rather than language (Kvale, 2007) although analysis of language also provided some insight into the views of the project managers. Analysing the meaning involved coding and interpreting what project managers said about ethnic diversity and the factors that influenced productivity. Coding frequency was used to help develop themes about what project managers identified as being most important.

Quantitative Analysis

The goal of the quantitative analysis was to determine the major productivity factors in the sample of software projects studied and the influence of ethnic diversity. One of the best ways of identifying those factors was to measure the association between variables and the strength of those associations. Correlation analysis was undertaken on the 34 variables in the conceptual model in order to help identify relationships that were potentially significant for productivity. Non-parametric tests were used as these are most appropriate for the type of data collected. These tests suit smaller samples (Hill & Lewicki, 2006; Siegel & Castellan, 1988) and have been used effectively with samples as small as five (Sheskin, 2007). Furthermore, parametric methods of correlation analysis (such as Pearson’s (Pearson, 1895)) are not suitable where one or both variables are categorical or ordinal (Green & Salkind, 2011) as some of the variables in this study are. Parametric methods were also not used as it was not possible to prove with confidence that all the prerequisite assumptions required were met (Coolican, 2009). The first assumption is that the data should be drawn from near normal distributions. This is difficult to assess with the data collected, so it is not possible to confidently make this assumption. The second assumption is that there should not be outliers, which could be errors. It is difficult to tell whether there are outliers in the sample being analysed. The third assumption is that there should not be heteroscedasticity. That is, the variance of the residuals should be similar, and this could not be proved for all variables.

Two non-parametric methods of correlation analysis (Spearman's rho and Kendall's Tau-b) were considered. Spearman's correlation ranks data and tests whether, if one variable for a project is ranked lowest, then the second variable being considered for correlation is also ranked lowest (Griffiths, 1980; Norušis, 2008). This analysis is useful for comparing data about the software projects as it indicates, for example, whether the most productive projects also had the highest diversity. Kendall's tau coefficient is also based on ranking items and is a relatively simple ranking based correlation compared to Spearman's correlation (Noether, 1981) but both are useful and accepted measures of correlation (Fagin, Kumar, & Sivakumar, 2003). The two non-parametric correlation techniques result in slightly differing meanings (Sprent & Smeeton, 2007) and each method "has its peculiar sensitivities and blind spots" (Hill & Lewicki, 2006, p. 385). Spearman's technique produces a product-moment correlation co-efficient whereas Kendall's technique produces a probability that the paired values (X and Y) are in the same order (Siegel & Castellan, 1988).

Kendall's technique is best suited to data where there are likely to be tied ranks. This occurs when a variable has only a few possible values (for example, low, medium and high) and a number of cases can fall into each of these categories (Malhotra & Birks, 2007). Spearman on the other hand is best when the variables have a large number of possible values, (for example, 1 - 100) and therefore there are fewer ties (Malhotra & Birks, 2007). There is mixture of these two types of variables in the data gathered in this study. For example, the variables which use a Likert-type scale would be better analysed using Kendall's technique as there are only a few possible values. However, other variables like productivity and project size have a large range of possible values and would therefore suit Spearman's technique. On balance, there are more variables with only a few possible values and therefore Kendall's method has been selected as the most suitable method of ranked correlation analysis. Only relationships that were significant under Kendall's method have been considered and investigated further.

Synthesis of Qualitative and Quantitative Data

The synthesis of the qualitative and quantitative results seeks to uncover additional findings by combining the results, over and above those findings identified in the two individual components. Venkatesh et al. emphasise the importance of this in their statement "the overarching goal of developing meta-inferences is to go beyond the findings from each study and develop an in-depth theoretical understanding that a single study cannot offer: a substantive theory of a phenomenon of interest" (2013, p. 19). Accordingly, the synthesis brings together the different facets of the research topic by presenting the dominant themes identified in the data analysis, bringing both the qualitative and quantitative findings together to build a more complete picture. The approach of bringing together the common themes but from the different perspectives is one effective way of ensuring the meta-inferences are richer than simply the sum of the qualitative and quantitative results (Bryman, 2004; Mingers, 2001; D. Morgan, 2007).

The qualitative and quantitative data collected on the 19 software projects was analysed using the complementary strengths of the different data collection methods (Greene, et al., 1989). Critical to a mixed methods study such as this one is the effective combining of qualitative and quantitative components (Morse & Niehaus, 2009). There are challenges in achieving this amalgamation because the two methods are grounded in different and arguably incompatible paradigms (Greene, 2007). At the core of this study are the qualitative data and methods which are supplemented by the quantitative components. Morse and Niehaus (2009) point out that the two points where the core and supplementary components meet in a mixed methods study are in the analysis and results.

Two broad approaches to deriving meaning from synthesising qualitative and quantitative results are bracketing and bridging (M. W. Lewis & Grimes, 1999). Bracketing enables analysing and drawing conclusions from disparate findings while bridging focuses more on identifying consensus between the two perspectives (Venkatesh, et al., 2013). The approach adopted for synthesis in this study is one of bridging, that is, focussing on themes and findings where both methods lead to either convergent findings or a broader understanding of relationships identified.

Conclusion

This chapter has presented the design and methods selected for this research and the justification for those selections. The theoretical framework has been outlined, including the relevant theories and the conceptual model. The sample selected for this study has been described, including a summary of the software development projects investigated. A description of the instruments used in this study, the data collection process and an explanation of how validity, legitimisation and transferability have been addressed in this study then follows. An overview of the ethical considerations addressed has been described, along with the pretesting that was performed. Finally an explanation has been provided of the analysis approach used to bring together the results. The next chapter presents the qualitative results arising from the interviews.

CHAPTER 4 – RESULTS AND ANALYSIS OF INTERVIEWS

Introduction

This chapter presents the qualitative results, reporting on the major factors affecting productivity and the influence of ethnic diversity. Those factors that influenced productivity within the ethnically diverse software development teams are analysed, beginning with the most frequently cited factors. The influence of ethnic diversity on each of the major productivity factors is also discussed where there was evidence that ethnic diversity was important. Factors that altered the influence of ethnic diversity are explored and the chapter concludes with a section examining the latent or underlying themes implicit in the project managers' responses. Where examples are provided from the interviews, the ethnicities in each project team are listed as this shows the degree to which the teams were ethnically diverse.

Communication

Communication was considered the most important factor affecting software development productivity. This was also the factor most influenced by ethnic diversity, consistent with earlier findings by the Diversity Research Network (Kochan, et al., 2003). The aspects of communication that were identified as being critical to the productivity of the software development projects are first discussed, followed by an explanation of how ethnic diversity within the teams influenced communication which in turn affected productivity.

Timeliness, the mode of communication and the use of feedback were the major communication issues. The key measure of good or effective communication was whether the correct message was received. Some of the communication issues were within project while others related to "boundary spanning" exchanges (Modaff, et al., 2008, p. 77) with stakeholders, other teams customers, vendors and project sponsors.

Within Team Communication

Language challenges

New Zealand is an English speaking country and the main effect of ethnic diversity identified by project managers related to problems with English. For example Jim (project 13) commented on communication challenges for one Sri Lankan team member who was new to the country and for whom English was not his first language:

> He's only been in the country for about 12 months. So his English is not as strong as the others. And of course with development work and when you're working on exciting projects and you get excited and you start to talk faster and faster it's harder and harder to understand. So we'd have to say, hang on, just slow down. And that was fine.

The project had a wide variety of ethnicities with Filipinos, Chinese, German, Latin American, Fijian / Indian, Afrikaner and New Zealand European team members. While Jim did not see ethnic diversity impacting either negatively or positively on productivity, he did take measures to facilitate team communication and develop team relationships. He discussed having informal stand up meetings each morning, as well as a more formal weekly meeting and once a month he would take the team out for coffee. These frequent and varied forums for communication within the team enabled effective information exchanges. The different types of forums enabled team members to communicate in the form with which they were most comfortable.

Isabel's project team (project 14) had eight different ethnic groups (German, Afrikaner, Fijian Indian, British, Latin American, Chinese, South East Asian and New Zealand European). She considered that while English as a second language was not a problem, it did create its own issues mainly concerned with comprehension:

> Fred can’t stand talking on phones because of the language. In Bob's case, he's been in the organisation for about three years, but I found his lack of English quite extreme. Not so bad in written form, but in verbal form. That certainly made a difference. Fred on the other hand, his English isn’t as good, but the only time that tends to bubble over is when he’s writing design documents – a big struggle. And he won’t use the phone. Which having worked in a foreign language I can understand to be honest.

Isabel's quote provides three examples of communication issues arising from ethnic diversity in the team; written documentation, general face-to-face discussions and talking on the phone. Isabel recognised these problems and managed the team accordingly. She later commented that team building, leadership and balance were important aspects of ensuring the different groups worked together productively.

Communication barriers between groups

Flora (project 19) talked about challenges where English was not a first language for some team members and they had difficulty with written and verbal communication. She commented that this was an issue for both the Afrikaners and the Indians in her project team. Although Flora described the problem as language related, some of her subsequent comments indicated that the communication challenges may relate to barriers between people seen as in-group or out-of-group. For example, she discussed overcoming the communication challenges by talking to the team about the need to work together to improve their communication with one another. She also discussed actions to help the team gel such as having them physically located together. The inclination for team members to be more likely to acknowledge and consider communication from in-group members (in this case, of the same ethnicity) rather than out-of-group members is considered to be a negative relationship related consequence of team diversity (Mackie, et al., 1990; van Knippenberg, et al., 2004). While this communication problem highlighted by Flora may, on face value, appear to be simply accents and aptitude with English, according to the social categorisation perspective it is likely to be underpinned by intergroup differences (van Knippenberg, 2007).

Communication barriers within groups

Communication was also hindered on Flora's project by hierarchies within the Indian subgroup where those of a lower caste would not speak up. Instead they deferred to the Indians in the group of a higher caste – in this case Brahmin. Flora described having come across a similar situation while working for a defence organisation where low ranking officers would defer to higher ranking officers when asked for input. She commented that “any sort of hierarchical issue is a problem in a project because everybody should have the same voice – and should be supported to say what they think”. These hierarchical issues represent a form of social categorisation. The observations of Flora are consistent with previous work which finds negative impacts of diversity arising from social categorisation (Dahlin, et al., 2005; van Knippenberg, et al., 2011).

Customer Communications

Communicating through documentation

The term ‘customer’ was used to describe both the end users of the software and those paying for the system. In the outsourced software projects customers were paying for the software developed by the vendors. For the in-house projects, a project sponsor or business owner was notionally paying for the software through a funding transfer internal to the organisation. The project sponsor was a role referred to in the in-house software development projects (as opposed to outsourced) and was the person ultimately responsible for the project results. Regardless of the nature of the customer relationship, communication between the customer and developer focussed primarily on articulating the software requirements and what the software developers must produce. Changes to these requirements during the software development process introduces additional rework and effort, thereby negatively affecting productivity (Boehm, et al., 2000b; C. Jones, 2008; Osmundson, et al., 2003). The more effectively a project team were able to be understood and thus meet customer needs without requirements change, the more productive a team was.

Some of the projects analysed were constrained by the customers' ability to communicate their requirements. For example Larry, the project manager for project 2 which had New Zealand European and Chinese Cambodian team members, commented:

> We would deliver something and they would say 'That's not quite how we kind of... now that we think about… maybe it should work like this. What do you think?

Larry believed the customer's uncertainty and inability to communicate what they wanted as the major factor negatively influencing his team's productivity. This was an outsourced project and Larry's software development team was not involved in undertaking the needs analysis and documenting the requirements. In this situation, Larry had to balance keeping the customer happy while avoiding making endless changes to the software as the customer's understanding of their own requirements evolved. While a requirements specification was the primary method of communicating what was required of the software, this was not sufficient to ensure effective customer communications. As Larry explained:

> What we've got in the specification is what we delivered and these guys are still trying to argue about whether this is the right thing to do.

In this project, the focus of the customer communication was on the limitations of using only a specification document but in other projects, project managers discussed how customer communication was enhanced through other approaches to communication.

Face-to-face communication

The use of face-to-face communication with the customer being physically present was the major method for improving productivity according to both Jim and Nick. Nick (project 9 with New Zealand European and Malaysian team members) explained:

> The key factor here in terms of productivity is that when there was an issue the business analyst, the developer and one or two of the user representatives actually got in a room and actually dealt with things. And that to me makes all the difference. Instead of emailing that person, back forward, back forward.

Here the customer communication was a transactional process (Miller, 2004) which was complex. The use of face-to-face feedback enabled the customer to clarify and seek other options and allowed the vendor to explain them. It also enabled the vendor to ask whether that is what the customer (in this case the end users) wanted. With frequent feedback to validate whether the message being received was correct, productivity was improved. Nick provided other examples from his project where face-to-face communication helped productivity. The underlying reason for favouring face-to-face was that it provided synchronous communication allowing ideas to be challenged, clarified and validated. This contrasts with forms of communication used on many of the projects such as email and documentation which is asynchronous and does not allow the immediate feedback.

Having a customer representative physically in the room with the developers each day enhanced communication and led to improved productivity. On project 13, which included Maori, Brazilian, Sri Lankan, Filipino, German and Irish team members, Jim explained "The business advisor was in there standing beside them each day. Asking them how’s this going, how’s that going.". This meant that the customers could identify at an early stage whether or not the system would meet their needs and the developers had a clear idea of what the customer wanted. This example highlights the ongoing nature of feedback to validate the message being received by the software developers and positions communication with customers as a transaction with the team "constantly participating in the communication activity" (Burgoon, Hunsaker, & Dawson, 1994, p. 13).

The benefit of using frequent feedback to enhance customer communication was also identified by Larry as being the major factor improving productivity on project 5 (which had New Zealand European and Chinese team members). He described how this worked by saying:

> They’ve said this – what are they actually wanting to achieve? Let’s give them a ring. Check with them. So removing ambiguity and just finessing things.

It was important that the project team sought to develop an empathy with the users of the system to improve communication. Larry explained how the team was "trying to get into their [the users'] shoes" in order to improve their understanding of what the customer required the software to do. The importance of empathy, face-to-face contact and frequent feedback were all identified as important factors in customer communication which affected the productivity of the software projects analysed.

Customer perceptions

Project managers reported that a diverse team appeared to the customer less likely to have an agenda or bias. The variety of backgrounds, worldviews and therefore perspectives meant decisions were more likely to be balanced and perceived by the customer as less likely to be skewed. Jim (project 13) reflected:

> I just think the customer found it very interesting that we had a project team made up of all nationalities. So there was no fixed agenda. We were doing this work for the customer and there was one European born in Germany, one European born in the UK, there was one person born in the Philippines, one in Sri Lanka. So there was nobody coming in with any set

> agenda. So for the customer that gave them a level of comfort as well that there was no issues [with individual bias].

While Jim did not directly associate a perceived lack of bias to productivity, he was clear that this helped the project deliver successfully by improving relations with the customer.

Vendor Communications

Communication between vendors working on different components of the overall software system is another type of boundary spanning communication which affected productivity. A lack of formal documentation to communicate how different systems would interface together was a problem. A typical example was on another one of Larry's projects (project 6) on which he identified this lack of communication between vendors as a major factor that negatively affected productivity. He described how "the supply chain is not managed by one vendor or one application and the touch points between vendors and applications have been fraught", by which he meant that suppliers that needed to work together were not communicating effectively. This caused a problem on the project because his team had responsibility for only one part of it and issues that arose from other sources came as a complete surprise. As a result of this occurring, the project team was forced to continually make changes to the software they were developing when the interfaces with components developed by other vendors failed.

Communication with Project Sponsor

Communication with the project’s sponsor or steering committee also proved to be a factor affecting productivity. The project sponsor is the individual in the organisation who is ultimately responsible for the project and who has the financial authority to pay the cost of the project. For large projects, there is a steering committee of senior managers who have a major stake in the success of the project, which is chaired by the sponsor. In one example (project 16) there was a lack of understanding between the project manager and the sponsor about whether or not the project was finished. Ralph explained:

> Everybody thought the project was finished, no one thought the changes were going to come in. So when the changes were communicated to us, all of a sudden we had to scamper for resources again.

Because Ralph understood the software project to be complete, when the sponsor informed him that there were more changes required, Ralph had to quickly find suitable skilled people to complete the changes. The original team members were no longer available so people who were unfamiliar with the software systems being developed had to be used instead.

Flora (project 19) indicated that early communication with the project sponsor, in this case the Chief Executive, improved productivity by ensuring the required resources could be obtained when required. When asked about the major factors affecting productivity, she replied

> Communication – at all levels. When I took the role, the first thing I did is ask the Chief Executive are you serious about this. And what I mean by that is when you do a project for an organisation that's disruptive, it's expensive and you have to make sure that three months down the road you're not going to say well actually we're too busy to give you a subject matter expert, or we're too busy for this, so it's commitment, organisational commitment.

By communicating what was needed to successfully deliver the project, she highlighted the need for commitment. Flora also went on to attribute communication with other stakeholders as being important to productivity on the software project. These stakeholders included users, customers and other business units affected by the project. The common theme highlighted is that the project managers perceived communication with key stakeholders outside of the project team as being pivotal to a productive project.

Communication, Ethnic Diversity and Project Size

In larger teams with many more potential communication paths, communication challenges were greater. Furthermore, complex software projects had a greater requirement for communication, collaboration and co-operation to solve problems and devise solutions. This increased volume and complexity of interactions had the potential to amplify the effect of communication barriers, language challenges and incompatible ways of working that can all inhibit working together. Therefore, the interviews have been analysed to consider how ethnic diversity on the large projects affected communication, given that communication was identified by the project managers as the primary factor affecting productivity.

Communication and team size

In order to examine the relationship between communication, team size and ethnic diversity, a clear definition of a large team is required. As the median number of team members was nine, for the purpose of this analysis, those teams with more than nine people are considered to be larger teams. Language challenges, ways of working that differed between team members and the additional time required to communicate across large teams were the issues identified by project managers. On the largest project, the project manager described how he spent a significant amount of time on teleconferences with the various groups within his wider project team as they were located all around the country. People would dial into phone conferences from different locations and this consumed a large amount of time. Geographic dispersion rather than specifically team size was the obvious issue. However, in order to resource a large project of 27 team members with the different skill sets required, it was necessary to engage people from different locations as they were not all available in the project’s base location.

In the larger teams there was greater team diversity in the widest sense of the term. This included different ethnicities, different ways of working and different employment arrangements, all leading to communication challenges. Project managers described ethnic diversity related problems where some team members had difficulties with English language affecting both written and verbal communication. In addition, different ways of working negatively affected the quality of documentation created. Other communication barriers reported on large projects were between contractors and permanent staff members. In contrast, the project managers of the smaller projects did not report communication problems.

Those projects with larger teams used a combination of formal weekly status update meetings, daily or ad hoc informal face-to-face communication and computer based knowledge management tools for communication. Email was used as a key communication tool in all projects. One large project adopted the agile software development practice of using a physical board with post-it notes to communicate the status of each component being developed. On the largest projects (with 22 and 27 team members) the project managers used a number of communication channels including daily five minute stand up meetings, formal weekly status update meetings as well as software tools for document, risk and issue management. There was also discussion of the importance of having the project team sit together to enhance communication within the team.

Project managers of large projects were required to focus on communication and this required additional hours of effort to be spent on the project, leading to poor productivity. In order to resource large projects, the project manager was required to source a wide range of people, from different organisations (such as a vendor), from different locations and from a wide variety of backgrounds. The differences inherent in larger teams increased the occurrence of communication challenges, such as language barriers and different work cultures.

Communication and project length

Long projects experienced challenges which negatively impacted productivity but their communication challenges were different to those generated by large teams. Six projects lasted multiple years and in that time, members of the original team left and new people joined. Asynchronous communication media such as documentation and knowledge management tools were used in order to track decisions made and information gathered. This would probably have been necessary even if there was no staff turnover in the team as people may forget what was agreed over a year ago. Productivity was impaired when team members had to re-litigate decisions or re-discover information.

Given that the longer projects had worse productivity than short projects, the communication aspects relating to project length from the qualitative data were examined. The longest project took two years from requirements gathering through to implementing the software. A number of changes in personnel occurred, both on the project team and in the customer teams. With the changes in personnel, good quality documentation was important. This also helped create a shared understanding across the project teams. However, the quality of documentation could be hindered by language challenges. In another long project, sub-teams came together at the beginning in the requirements phase, then dispersed for many months to work on their respective components before coming together again at the end to integrate the parts together. It was in this integration stage at the end that the quality of the communication throughout the project was tested. At the point of integration, the components developed separately only worked together well where the communication between the separate teams had been effective. Also on long projects, techniques like daily status updates could not be maintained over a long period of time as team members just stopped attending. While some project managers found daily status updates effective with larger teams, this only worked for short periods. This is significant for effectively managing ethnically diverse software teams as daily stand up meetings were one way to encourage everyone to contribute and share their ideas or concerns. Given that daily meetings such as these cannot be sustained on long projects, other methods for creating a team environment where all perspectives can be heard become more important.

Communication and product complexity

The relationships between complexity, communication and productivity in the interviews reveal key findings regarding maximising the potential benefits of ethnic diversity in software teams. On the projects with high complexity, communication requirements increased due to multiple sub-teams working together, debates with the customer relating to product complexity and the need for team members to justify the decisions made in the development of a complex software product. On the most complex project there were multiple architecture layers or components to the system being enhanced. Each component was developed using a different technology and programming language, by a different team which specialised in that component's technology. This created a requirement to communicate across these teams, who all effectively spoke a different technical language and viewed the world slightly differently. In this way, technical complexity led to social complexity, reflecting the socio-technical nature of software development. Other projects which were rated as having high product complexity also involved multiple architectural components developed by different people and teams. This is an example of high complexity combining with communication challenges to negatively influence productivity.

Communication Differences between Government and Non-Government

Communication was rated as one of top three productivity factors by both government and non-government project managers. Although both groups of project managers identified communication as affecting productivity, government and non-government project managers focussed on different aspects of communication. Non-government project managers talked almost exclusively about boundary spanning communication which impacted on productivity including communication with customers and vendors. Customer communication included both end users of the software as well as those paying for the system. Regardless of the nature of the customer relationship, communication was focussed primarily on articulating what the software was required to do. Communication between vendors was a major factor affecting productivity on non-government projects where different vendors were working on different components of the overall software system. The issues identified related to a lack of formal documentation to communicate how different systems would interface together. By contrast, government project managers focussed on within system communication issues either amongst project team members or with the project sponsor. Some government project managers explained how positive internal communication resulted in a highly productive project team while others cited negative internal communication causing a less productive team.

Although communication barriers were discussed by government and non-government project managers, it was only in the government projects that the issues were associated with ethnic diversity. The main issue was some ethnic groups not speaking up in the way the project manager expected. In one project, the problem was related to those of a lower caste not speaking up in the team and instead deferring to those of a higher caste. In another case, the project manager considered it to be a cultural thing about people not always putting up their hand and admitting 'I don't understand' or 'I can't do it'. In these cases, the project managers linked unwillingness to speak up, to negative impacts on the productivity of the project. In contrast, the non-government project managers did not report any ethnicity related communication barriers on their projects.

Language challenges were discussed by four government project managers but only one non-government project manager. Three of the four government project managers linked the language challenges to ethnicity and described how one or more team members' poor English negatively impacted the project. By contrast, the one non-government project manager who reported language challenges on his project did not feel that this affected productivity. Despite one team member having some problems with English, his ability to perform his software development role was not affected and this was attributed to having the right people in the right roles.

Summary of Communication Aspects

Ethnic diversity within the teams was identified by some project managers as negatively affecting communication. Having different ethnicities in a project team led in some cases to problems with English as a second language making it more difficult to understand a person, their written documentation, face-to-face discussions and talking on the telephone. Linked closely with this are the cultural differences which could arise in certain circumstances and contribute to misunderstandings. One project manager also stated that having an ethnically diverse team improved interactions with the customer as the team was viewed by the customer as being balanced and less likely to be skewed to one specific view. While communication issues were discussed by project managers in relation to the effect of ethnic diversity within their teams, communication was also identified as an interrelated factor with other important themes. In particular, the team cohesion was both an enabler of effective communication and was enhanced by effective communication.

Team Cohesion

After communication, cohesion within the team was the second most widely cited factor affecting software development productivity. Although this term was not specifically used by project managers in responding to the open question about what affected productivity, the answers given in a number of cases relate to team cohesion. In most cases high cohesion was seen as improving productivity and in one project low cohesion was identified as negatively affecting productivity.

Two main areas of team cohesion were identified in the interviews. The first area was the need for the team to have shared goals and the second area was the strength of the relationships within the team. Consistent with social exchange theory (Blau, 1964; Cropanzano & Mitchell, 2005) the team relationships were underpinned by reciprocity and negotiated rules. The project managers reported that team cohesion was in some situations helped by ethnic diversity in the team, and in other situations hindered. Having different backgrounds generated interest amongst team members and acted as a catalyst for initiating and maintaining relationships between team members. However, the different ways of working associated with ethnic diversity had some negative effects on cohesion.

Shared Goals

The alignment of the teams' goals was identified by project managers as an important way to ensure the whole team was working towards the correct objectives. Where the team had a common understanding of what outcomes were required from the project and were all working towards those outcomes, productivity was improved. Ensuring all team members were working towards the correct project goals was described as both a positive outcome arising from team cohesion as well as a means of achieving team cohesion. In four of the projects, there was discussion about the value of common goals in achieving team cohesion and improving productivity. Ralph (project 16) commented that "the team was gelled" and described his team which had Indians, Asians and New Zealanders as "close knit". He accounted for this team cohesion in terms of a clear understanding of the project's objectives and a high level of buy-in to those objectives. He explained:

> From the start...everybody understood what was the reason for the project, which was the legislation, and everybody understood what was going to be required out of the system.

The shared goals supported not only cohesiveness within the team, but also clarity on the end product being produced. By having everyone in the team clear about the purpose of the project and what was required from the team, they were better able to produce the required output. Ralph felt there were no negative effects from the diversity within the team and that this was because everyone was focused on the results and openly communicating to get the job done.

This focus on the clarity and visibility of the project's objectives was also identified by Karl on project 17 which included Indian, Irish, American and New Zealand European team members. He commented, "There's been a lot of commitment from the team. They've taken ownership of building something that they're proud of." Karl attributed the team cohesion to the project goals being made highly visible through the use of post-it notes on a board visible to whole team. This way, the team could clearly see what each other were working on and the progress of interdependent components. Ensuring that team members could clearly see the project status and what needed to be delivered, helped the team work cohesively together.

In contrast, in project 15 a lack of common, clearly understood project goals was described as an important factor that led to poor productivity. The project included Indians, Asians and New

Zealanders and was described by Ralph as being relatively unproductive. The reason for this was that the development team attempted to document how the entire system worked due to a lack of existing system documentation. This activity absorbed much of the time allocated to the project, and as this was not the purpose of the project it caused major problems. A delay of three months resulted and the project incurred additional unplanned costs. In this example, the misalignment of goals between the development team members and the project manager had a negative effect on the project's productivity.

Use of Feedback

One method that project managers used to help align team goals with the outcomes required was the use of frequent feedback. This took two forms. One was feedback provided to the team regarding whether they were building the solution correctly. The second was the team themselves providing feedback to their customers about what they understood was required.

In project 9, Nick described taking the customer to see the vendor to evaluate early versions of the software. "Taking the customer to the [developers] was a good example. They would go for a first look and could say 'oh yes that looks okay' or 'could you make these adjustments?'." The physical presence of developers and customers in the same location allowed feedback to be given verbally in a face-to-face manner. This had the additional advantage of the developers seeing the customers as people and helped to personalise feedback, increase its effectiveness in focussing the developers on the required outcomes and develop a relationship. Such face-to-face discussion between the developers and the customer enabled richer and more complex communication to occur.

Another method of obtaining feedback to help align team goals with required outcomes was having the customer representative work with the development team every day. Jim (project 13) explained that this resulted in the developers knowing "that they weren’t working in isolation – they were part of a team that was wanting to make it all happen together". In this example, the feedback is not just face-to-face, it is also frequent. The daily checking and feeding back to developers further kept the team focussed on the project goals.

This was extended further on project 17 where there was not just continuous feedback from the test team to the developers, but also "a real buy in between the developers and testers". The ethnicities included in the team were Irish, Indian, American and New Zealand European. Karl, the project manager, attributed the team's productivity to good team cohesion arising from the frequent feedback that was facilitated, in part, through a collaborative environment where everyone was involved in design and development decisions.

Strong Team Relationships

The second key aspect of team cohesion was the importance of developing familiarity among team members through measures such as team building activities and seating the whole team together. While most project managers implied that sitting together as a team was important, the following quote from Flora (project 16) shows her focus on the importance of having the project team physically situated in the same place:

> The project team all sat together. So you’ve got to be part of a physical team. And out of that comes, everyone can see what everyone else is doing, and they can all support each other. It makes a huge difference if they all sit together. I went to a lot of trouble to get them all to sit together – even people who were from suppliers.

Having the team in close proximity to one another facilitated immediate reciprocity which supported the development of relationships between team members. A number of project managers sought to build familiarity among team members to enhance cohesion through team building. For example, Mark (project 18) commented "there were a reasonable number of social get togethers, lunches and dinners and things". Social interaction improved relationships between members and contributes to the development of trust and confidence in one another.

Another aspect of social exchange theory that project managers highlighted in their responses on building team cohesion was that of negotiated rules (Molm, 2003). Mark (project 18) described his project team as productive and explained how negotiated rules worked in his team. "There was a working together charter put together at the vendor's initiative. The management of both organisations went to great pains to ensure we were working in a true partnership". The team of 22 was drawn from both the organisation for which the system was developed and the vendor, and the charter formalised the agreement on how the partnership would operate in the form of negotiated rules.

Not all projects enjoyed the benefits of good team cohesion and in one case this was caused by poor relationships between team members. Ralph described the team on project 15 as unproductive and attributed poor team cohesion to two causes. The first was lack of opportunity to establish team relationships at the beginning of the project which arose from the team being engaged late and not being involved in the planning. He discussed how there was "no establishment of the real team beforehand. So it was like 'this is it, you do the work, report back to me"' (Ralph). The second factor that undermined team cohesion on project 15 was friction within the team. He explained why this occurred and how it affected the project:

> Because they were contractors they were being treated by their core team mates as 'you do the bulk of the work because you're the contractor. I'm the permanent I'm just doing this'. It's like 'I'm not telling you stuff so you do it on your own, you have to discover it.' Basically – just a lot of misunderstanding – even to the point of animosity.

The lack of team cohesion in this project appears to be brought about in part through resentment by permanent employees towards contractors. This situation of mixed contractors and permanent software teams has been found previously to lead to decreased team cohesion (Ang & Slaughter, 2001). In this situation, early establishment of team relationships in order to build either negotiated rules or nurture reciprocity amongst team members is likely to have improved team cohesion.

The Influence of Ethnic Diversity on Cohesion

Some project managers stated that ethnic diversity improved team cohesion arising from both task and relationship related aspects. Because of the variety of backgrounds and experiences associated with ethnic diversity, team members were interested in finding out about each other and this acted as a catalyst for initiating and maintaining relationships between them, particularly in informal settings. The potential for ethnic diversity to improve relationships was most significant where an environment of trust was created within the project by the project manager. Opportunities for relationship building needed to occur or be created where interest in other team members' background and experiences could be explored and used to strengthen team connections. In this way the project manager played a key role in facilitating relationships in the project teams and especially so in ethnically diverse teams where there were higher risks of social categorisation.

Something interesting to talk about

Working with people from different backgrounds and perspectives helped to keep the team members engaged. The variety of backgrounds and experiences that was associated with ethnic diversity in the team led to a greater level of interest in finding out more about each other and therefore forming stronger relationships. In this way, ethnic diversity was seen to improve team productivity by helping to build effective work relationships within the team. This effect is explained by Karl when he described how the team's ethnic diversity affected its productivity:

> I think positively. You definitely do not want everyone to be the same. Not only is it boring but it doesn't give you stuff to talk about at the pub if you're not having a little polite jab at each other with everyone in attendance. There's nothing better than there being a little bit of banter about your background or how you do your work. So I think that's really positive. People are learning from each other.

Team members treated the fact that others were different from themselves as a positive aspect. Generally, ethnic diversity has been considered to have a negative effect on relationships due to individuals seeing dissimilar others as outside of their group, leading to communications challenges and conflict due to disparate values (Ayub & Jehn, 2011; Jehn, et al., 1999; Sujin, 2005). Therefore, the positive effect on relationships that occurred on this project differs to previous findings on team diversity.

Different ways of working

Team members from different ethnic backgrounds had different ways of working and this had both positive and negative effects on team cohesion. Conflict arose when working styles clashed. For example, Flora recounted how two South Africans on her team had work processes which did not work well with the rest of the team. She recounted how she "had problems with both of them with their work process." Flora went on to explain:

> The team worked together to quality assure each other's (work) and one of the South African women in particular hadn't been through that process so she had much less rigour and it had an impact on all the other members of the team because her work wasn't up to scratch.

Although the different working styles were linked to ethnicity, it was not explicitly attributed to it. Having team members with different ways of working was also seen as positive in some contexts. One of the benefits was that people could be assigned to a role that suited their work style as there are a number of different roles on software projects. For example, Isabel commented:

> I think people put themselves into the role that matches their personality and ethnicity. So for example, software development tends to be something that isn't as sociable or as outgoing. If you're a project manager, you've got to be out there and you've got to be arguing.

Isabel went on to explain how having the right people in the right roles, combined with what she called "standard team building and your leadership" helped to form "levels of grouping within the teams". Although she did not attribute the different ways of working solely to ethnicity, her responses showed that she felt it played a role. A similar sentiment was raised by Larry, although he couldn't say for sure that it was related to ethnicity. He explained how various team members had the right working style for their respective roles and it was important with this small team to have a variety of complementary working styles. He finished by discussing how a Cambodian team member was the right person for the user acceptance testing role due to his level of patience and commented "Is that part of his ethnicity? I don't know. Could be".

Co-located teams

Cultural differences in Flora's project (19) gave rise to conflict between team members regarding how certain software development tasks should be done and she described her strategy for addressing the problem:

> There are cultural differences that affect productivity. And it's just getting used to working with people. If you sit them all together and provide strong leadership then you all have a goal. Everybody knows everybody else's timetable and the barriers start to come down. We also used to have quite a bit of social downtime when they weren't having to produce stuff. They were getting to know each other. That brought down barriers too.

This wasn't the only time that Flora discussed the benefits of having all of her project team seated together, but here it was specifically focussed on dealing with challenges associated with cultural differences. She and other project managers (such as Karl on project 17) focused on building relationships between team members to deal with any potential negative effects of ethnic diversity. They showed that developing these social connections helped overcome the negative impact of social categorisation in diverse teams described by van Knippenberg et al. (2011), or relationship conflict which Liang et al. (2009) attributed to diversity within software teams.

Trust

As an ethnically diverse team requires individuals to interact across cultural boundaries, relationships can be weakened (Goodwin, 1999) due to communication barriers, conflicting values and social categorisation. This makes it essential that the project manager establishes an environment of trust in ethnically diverse teams. In the projects where project managers described how trust was present in the team, the benefits of team diversity were more likely to occur.

While some project managers explicitly discussed how trust existed within the team, others clearly implied the need for trust in their responses. Trust enabled team members to ask questions without fear. Jim (project 13) commented that "we had the level of trust within the team. We weren’t backward in coming forward in actually saying “what did you say?” “what do you mean by that?”. While other project managers did not use the word trust, their responses indicated situations where trust enabled open communication. For example, Karl (project 17) noted "There’s a real buy in between the developers and testers and the business analysts and the project managers. A real open forum which is unique for this place.” The theme was also raised by Ralph (project 16) when he explained that team diversity helped make his project productive because "everybody was just trying to communicate really and asking questions no matter how dumb the questions were. Which I think was positive. Everybody was just asking questions.” Ralph considered that team members were only going ask to questions if they trusted other team members not to ridicule or ostracise them.

All three of these project managers identified their team as being ethnically diverse and also described their projects as successful. They believed their projects benefited from having diverse perspectives on the software development tasks undertaken by their teams. Without trust, these diverse perspectives are less likely to have been raised and discussed. Trust was a critical enabler for effective communications within the project teams as well as being important for achieving team cohesion. Furthermore, the ability to improve relationships between team members was enhanced where an environment of trust was created within the project by the project manager.

Team Cohesion Differences between Government and Non-Government

Team cohesion was viewed by government project managers as being far more critical to software development productivity than by non-government project managers. Most government project managers described how a highly cohesive team improved productivity although one project manager discussed low team cohesion negatively impacting productivity.

In four of the government projects, there was discussion of the value of common goals in achieving team cohesion and improving productivity. By ensuring that team members could clearly see the project status and what needed to be delivered, this helped ensure the team was working cohesively. One government project manager attributed the team's good productivity to good team cohesion arising from the frequent feedback that was facilitated in part through a collaborative team environment. The second key aspect of team cohesion which was evident in the interviews with government project managers was the importance of developing familiarity among team members through measures such as team building activities and seating the project team together. Another aspect that government project managers highlighted in their responses on building team cohesion was that of negotiated rules (Molm, 2003). Not all government projects enjoyed the benefits of good team cohesion and this was attributed to two causes. The first was a lack of team building arising from the team being engaged late and not being involved in the planning. The second factor that undermined team cohesion was friction within the team brought about in part through resentment by permanent employees towards contractors.

The non-government project managers who raised team cohesion discussed how good team cohesion on their projects was a major positive factor affecting productivity. Two of the project managers identified the importance of aligning team goals to the outcomes required. The third non-government project manager identified team cohesion as an important factor, describing how his project team had been together for some time and therefore worked together well. In this case, familiarity between team members led to strong relationships within the team. While both government and non-government project managers discussed the value of shared goals, there was more focus and effort in the government projects on developing relationships within the team through team building and seating the team together in the same area.

In both government and non-government projects, ethnic diversity was associated with having team members who worked in different ways. Some government project managers saw ethnic diversity as having a negative effect on team cohesion. Their perception was that people from a different country worked differently and this hindered the team working together effectively. In contrast, on non-government projects the project managers described how team members had different but complementary work styles. The different roles required diverse working styles and it was seen as an advantage to have people with different ways of working.

Requirements

After communication and team cohesion, requirements aspects were the third most frequently cited factor affecting productivity. Requirements quality and requirements change were the two main issues where requirements affected productivity. None of the project managers indicated that ethnic diversity altered how requirements quality and volatility affected productivity although other major factors affecting productivity are interrelated with the requirements aspects. Communication affects how effectively requirements are conveyed to the software development team and whether developers are able to ask questions to clarify their understanding of the software required. Team cohesion also affected how well a team developed accurate requirements and dealt with any changes to those requirements.

Requirements Quality

Requirements quality was raised in a number of interviews, with project managers commenting that poor quality requirements negatively influenced productivity. For example, in discussing project 6, Larry noted that poor quality requirements resulted in a large amount of rework. He identified the source of this problem as being the developers writing the requirements for the software rather than the end users. Larry went on to explain that because users of the software are the best people to specify the requirements, the software development team did not get the requirements right in all cases. This is because it is difficult for a software development team to deliver the correct software if those who are using the system do not explain or document what they require. As a result of this, many changes were required to software before it could be used and Larry described this project as "one of the least productive projects that I have been associated with in my years".

Another project where poor quality requirements negatively affected productivity was project 15, where the requirements were inaccurate and insufficiently documented. Ralph (the project manager) elucidated this by saying:

> On the inputs first, the requirements were really vague. And the requirements were even contradicting existing functionality, requirements were even contradicting the legislation, requirements were even contradicting operations in the field. The requirements were wrong from the start.

This affected productivity because, as Ralph describes it "there was a lot of wastage in terms of the code" as they had to rewrite software when it was discovered that it did not meet the business's needs. This was attributed to those developing the requirements lacking the necessary skills to effectively capture and document software requirements.

Other project managers described how good quality requirements improved productivity. When discussing project 1, Larry responded to the question ‘what impacted productivity?’ by explaining how good quality requirements were especially helpful when two people or more were involved in development at different times. Having good requirements enabled the work to be handed over from the first developer who was familiar with the project to the second developer without negatively impacting the project. This resulted in a very productive project.

Nick (project 9) also explained how high quality requirements were achieved during the requirements phase by having the developers in workshops with the users. This helped the developers understand the business processes and what the users were trying to accomplish. By having the developers in the user workshops, two positive outcomes were achieved. One is that the developers were able to ask questions during requirements gathering which related to the effectiveness of communication discussed in the previous section. This helped ensure the requirements did not leave important aspects undocumented. Secondly, the developers had a better understanding of the requirements as they understood the business process and goals. By having clear and accurate requirements, this project was able to deliver a system which met the customer needs in a way that the project manager described as productive.

Requirements Change

The second key reason for requirements impacting productivity was requirement change. This is referred to as requirement evolution and volatility in COCOMO II. The later in the project lifecycle changes are made to requirements, the more costly it is to include those changes (Chua & Verner, 2010). As a result, requirement changes (or volatility) negatively impacts productivity. In some cases the requirements were clear at the start, but the customers changed their minds about what they wanted when they received the software. Ralph (project 16) commented that productivity was going well until the customer requested changes. Changes were required by the business late in the project and eroded earlier productivity gains. In another project, the source of the changing requirements was evident from the start. Larry (project 2) discussed how the customer was uncertain about what they wanted throughout the project. This resulted in changes to the documented requirements once the system was delivered and the customer saw what they were getting.

However, requirements changes did not always impact negatively on productivity where they were managed effectively. In one of Larry's more successful projects (project 5), he described how he handled change in requirements. A request from the customer to make a major change to requirements was discussed but deferred to a later project. Although minor requirements were added, the major changes were successfully deferred after some negotiation. Managing requirements is an important component of the project management discipline (Hartley, 2008; Pinto, 2009; Project Management Institute, 2008) and Larry's discussion about project 5 is an example of effectively managing requirements change. However, in practice this is not always possible. For example where customer satisfaction is more important than delivering on time and within budget, it may be necessary for a project manager to accept a change to requirements, and the consequential negative impact on productivity (Meredith & Mantel, 2008, p. 502).

Team Capability

The Influence of Team Capability on Productivity

The capability of the team was discussed as a significant factor affecting the productivity achieved. As well as using the term 'capability', project managers also used closely related phrases such as "highly skilled" and "experienced" and "very senior". For the purpose of this analysis, these are discussed in this subsection under the general term 'team capability'. In most instances where these factors were raised by project managers, it related to technical capability, in either the developers or solution architects although in one case a project manager referred to the business analyst being "out of her depth" (Isabel). Four project managers attributed a productive project to having highly capable or experienced staff (projects 1, 9, 13 and 18). For example on project 13 Jim attributed good productivity to "having highly skilled staff". For two project managers, poor productivity was due, at least in part, to a lack of capability (projects 6 and 14). For example, Larry described how the lead developer’s lack of capability led to poor productivity on project 6.

The Influence of Ethnic Diversity on Capability

Ethnic diversity was seen by two of the project managers as helping the team perform by bringing different perspectives to the tasks being undertaken by the team. In order for those different perspectives to be communicated and therefore considered by the team, a team environment is required in which any and all questions can be raised without fear. Ralph (project 16) noted that he had a team environment which benefited from everyone being encouraged to ask any question, even if they thought it was a “dumb” question. His government project had a team of 11 with four ethnic groups represented (Chinese, Southeast Asian, Indian and New Zealand European) and when asked about how the ethnic diversity in his team affected the team environment, he replied "positively", explaining how everyone in the team was able to ask questions. When probed further regarding any negative aspects of diversity, he said “Negative impacts? I don’t think there was any kind of negativity caused by diversity.” (Ralph)

This benefit of bringing different perspectives to a task was also illustrated by Karl (project 17). His project team of 12 included four different ethnic groups (Indian, American, Irish and New Zealand European) and Karl described how design decisions were always made by three people:

> You don’t want to only go to one person for a solution. We have a rule that we never have only two people in a design discussion. There’s always got to be a third because we’ve had arguments over designs. There would have to be a third person that they have to challenge or coerce as to which design was best.

Karl described his project as successful and the productivity achieved was the best of the seven government projects analysed. This quote shows the propensity for differing and strongly held views within the ethnically diverse team. By allowing these views to be voiced and then distilled into an agreed design through debate, the best design solution was able to be arrived at. Karl's approach to achieving design decisions led to the "need to integrate and reconcile diverse perspectives" (van Knippenberg, 2007, p. 13), which has previously been shown to help stimulate creativity and innovation.

Team Capability and Ethnic Diversity in Government and Non-Government

Diverse perspectives were generally seen to enhance overall team capability in both government and non-government projects. Project managers from both sectors described how diverse perspectives helped in project teams where there was ethnic diversity. However, government project managers were more willing to discuss how ethnic diversity affected productivity on their software projects. When non-government projects did discuss ethnic diversity, it was always from a positive perspective, focussing on how productivity was assisted by diverse perspectives. Even language difficulties were not identified as a problem as the developer involved was in the right role where this did not cause a problem. Government project managers reported a mixture of positive and negative influences of ethnic diversity on their teams' productivity. In the negative cases, team members had conflicting ways of working or communication challenges. Given some of the comments from project managers about dealing with team challenges, there was some indication that non-government project managers had more control over the selection of team members.

Attitude and Motivation

The attitude and motivation of team members were also seen as key influencers of productivity in some software teams. The project managers of seven projects stated that having the right attitude and being motivated were important contributors to productivity (projects 4, 5, 6, 9, 14, 17 and 18). On project 17 Karl noted that the diverse roles to which people were assigned kept them motivated, such as working on new applications as well as maintaining existing ones. He went on to comment:

> I don’t think the organisation has really cottoned on to the fact that we need to be doing more to give people the diversity of roles to keep them interested.

Another perspective on attitude was described by Larry in project 6 where the organisation and its staff needed to change to being less sensitive to people related issues and instead focus on delivering profitable projects. Larry commented “so I’m afraid we’ve got a lot more hard-nosed – it’s like if you’re not performing, you’re out and gone”. He identified the importance of team members bringing the ‘right’ attitude to ensure a productive software project. While project 6 was a non-government project, a similar sentiment was discussed by Isabel (project 14) on her government project. She commented:

> You’ve got the ones who would buy in and be there at 8 pm at night and say we’ll fix this; and the other ones who would shrug and disappear off home. It would peeve people off.

However in contrast to the non-government project, people with the “wrong” attitude are not “out and gone”. Instead the project manager relied on dynamics that naturally formed within the team. Isabel commented:

> When you get the people who don’t [do their share], they tend to get more ostracised. Which is understandable. If you’re not doing your share I don’t want you on my team – sort of thing.

This project had a high degree of ethnic diversity with at least eight different ethnic groups in the team of 15 people. The differing attitudes to dealing with project challenges led to relationship conflict. Unlike task conflict which is considered to have potentially beneficial effects, relationship conflict is generally considered to be counter-productive (Ayub & Jehn, 2011; Horwitz, 2005; van Knippenberg, 2007). In Isabel's project, the relationship conflict was under-pinned by differing values, reflected in attitudes, and it is culture which influences one's “values, beliefs, norms, and behavioral patterns” (Leung, Bhagat, Buchan, Erez, & Gibson, 2005, p. 357). In the example of relationship conflict described by Isabel, values, norms and behaviours differed between team members regarding what was appropriate in a situation where the software project was having trouble and work needed to be done to solve a problem. The fact that some team members would stay behind and work to solve the project problem and others would "shrug and disappear off home" reflects differing levels of in-group collectivism, an important aspect of cultural values identified by Hofstede (1980a) and others (House, et al., 2002; Javidan, House, Dorfman, Hanges, & Luque, 2006). Those willing to stay behind for the good of the project exhibited a high level of in-group collectivism compared to those who simply left to go home. The quote from Isabel highlights a clash of the cultural value of individualism “where people take care of themselves” (Hofstede, 1980b, p. 45) versus collectivism “characterized by a tight social framework in which people … expect their group to look after them, and in exchange for that they feel they owe absolute loyalty to it” (p. 45). In Isabel’s project, this clash led to conflict between those team members who exhibited a high degree of loyalty to the project team and chose to stay behind, and those who don’t.

Use of Agile Practices

The use of agile development practices improved productivity and was discussed on seven of the projects. The use of agile practices was identified as improving communication and enhancing team cohesion, both of which improved productivity. Agile is an approach to software development that encourages face-to-face communication over documentation (Beck, et al., 2001; Fowler, 2005). There are a number of different agile software development methodologies, one of which is Scrum (Dybå & Dingsøyr, 2008). This methodology was adopted to varying degrees and affected communication in the software projects at organisations E and F. Scrum was used extensively in project 17 and some practices were adopted in projects 13 and 18. For example, in response to the question about the methodology use on project 13, Jim replied:

> Modified Waterfall / Agile. Because of the organisation we just couldn’t physically create the perfect agile environment where everyone is in one room, they’ve got glass walls all around them, they can stick labels all over the place – we didn’t have that. So I think we did a fit for purpose agile approach.

Further on in the interview, Jim elaborated on the use of daily stand up meetings for informal sharing of information and any challenges, complemented with more formal weekly meetings where minutes were taken and the issues and risk registers were updated.

Another aspect of the agile approach was the constant feedback to the team. This not only helped them deliver the software that the users wanted but also made them feel engaged. Jim explained that on project 10, the business analyst appreciated being involved with the project team every day as he was able to provide input on what the customer wanted and then received feedback from the team about what was and wasn't feasible. This approach resulted in the finished product being closer to what was expected with less opportunity for surprises about how the software operated. Jim commented that the adoption of agile practices made such a significant and positive impact on the success of the project that:

> The business don’t want to do any other projects any other way because for once they felt they had total involvement in the project and what was delivered was exactly what they asked for.

On project 17, Karl fully adopted an agile development approach. He also commented on the value of feedback in building team cohesion by saying "the productivity has been improved because the 'agile' approach has meant the test team and the developers have provided continuous feedback.”

Another benefit of the agile approach that the project manager for project 10 discussed was the use of an agile collaboration tool. An agile collaboration tool was used on projects 13 and 17 and specifically identified as an enabler for success by the two project managers for these projects. On project 17 they had a physical "scrum board" which showed the progress of each component being developed. This was complemented by an electronic repository which held detailed information about each change and its status. Karl described how this combination helped team cohesion and communication by explaining how they used the scrum board for daily scrums (regular short status update meetings). He then went on to discuss the electronic tool and commented "So we’ve found that is a real pro [benefit] – having a combination of our electronic repository of this information in the tool and having a physical board as well for the things that are immediate.”

In summary, where agile development practices were used, project managers described how this helped productivity, particularly in terms of communication, feedback and co-ordination. Communication was helped through daily stand-up meetings which are based on the principle of frequent face-to-face communication described in the agile manifesto (Beck, et al., 2001). Early and constant feedback to the development team from both the customer and the testers contributed to team cohesion and ensured the software developed met the customer's needs. Finally, the use of agile-oriented team collaboration software tools helped the team communicate and stay focussed on the highest priority requirements. The benefits of using agile practices are interrelated with the other important productivity factors of team cohesion and communication.

Complexity

The complexity of the software development projects was an important factor cited on four projects. The type of complexity related to complicated business requirements and interdependencies with other projects, rather than any technical complexity such as using new or complicated technology. In most cases where complexity was raised it negatively affected productivity. When discussing project 2, Larry commented how the project was productive, despite the complexity. When asked how he could tell the project was productive, he replied "when I look at the bewildering complexity of the rules, the fact that we managed to get anything out there that worked 99% of the time is astonishing to me." Complex business requirements were cited on projects 14 and 18 as having a negative influence on productivity. Isabel (project 14) said that "people didn't accept the complexity of it." and went on to describe how complicated the functionality required of the software was. This is because at a high level, the function of the system appeared simple, but when the functionality was examined in detail, as is required in order to develop software, it became clear to the project team there were a number of complex subtleties to the requirements rules.

The negative impact of the complex business requirements were compounded by the business analysts being "out of their depth". In order to understand and specify the software requirements the business analyst representing the business needed a detailed understanding of fuzzy logic concepts, which they did not have. Despite these challenges, the ethnically diverse team of 15 were able to successfully deliver the required software, albeit with relatively poor productivity overall.

The second aspect of complexity was interdependencies with other projects. Andy (project 11) explained how the organisation had between 300 or 400 software projects going at the same time. As with many large organisations, there were many systems with numerous interfaces between them. "Those projects inevitably become intertwined in terms of someone changing this system and that all created dependencies" (Andy). He went on to explain how this created a lot of complexity which negatively impacted the productivity of his project. The project team of 21 included Maori, Indian, South African, Pacific Island and New Zealand European team members. As the project team faced complex and challenging problems, Andy commented that the diversity of approaches, which was in part related to cultural background, helped them find solutions. What was initially viewed as a tension between two different design approaches was later seen as a benefit. Because one team member came at the problem from a quite different way from the project manager, different possibilities became apparent. This type of benefit of ethnic diversity relating to innovation and problem solving has been previously reported in studies into ethnically diverse teams (Post, et al., 2009; Tadmor, et al., 2012; Winkler & Bouncken, 2009).

Application Experience

Having one or more team members who had experience with the software application being changed or replaced was identified as improving productivity. On three projects, having application experience helped the team's productivity while on project 9, a lack of experience had a negative effect on productivity. Project 9 involved replacing an existing system and Nick the project manager explained how "initially there was a lack of understanding of what their existing system did." As a result, the developers and business analysts who were responsible for developing the replacement system "took a period of time to come up to speed to work out what they actually wanted the system to do" (Nick).

In contrast to project 9, where there was a lack of understanding of the end user functionality, the other three projects benefited from a good understanding of both end user functionality and the internal structure of application. This is because in all three of these projects, changes were required to an existing software system. On project 10, some of the team had been working on the application since it was first developed approximately 30 years prior to the current project. This high level of applications experience was cited as the main factor that contributed to a productive project. On projects 1 and 2, a good background in the application being changed helped produce accurate estimates and deliver a productive project.

Planning and Design

The benefits of early planning at the start of the project were highlighted by Ralph on project 16. In response to the question about what affected productivity on his project, he stated:

> Early planning...Because you had early planning, people began to ask questions early on. So before development even started, the development team was already asking what’s going to happen to this, what’s going to happen to that.

However, when discussing another project that he had managed (project 15), he commented on the negative impact resulting from a lack of planning. He explained:

> No planning was really done. Although there was planning as in establish the dates, establish the estimates, there was no real involving the business in terms of the requirements.

Larry also cited a lack of planning as a negative impact on productivity for project 6 where having a business analyst rather than a project manager at the start, resulted in no project plan being developed. When a plan was developed in project 18, Mark found effective and frequent updating of the project plan had a positive effect on productivity. He explained how he elicited ongoing, accurate estimate revisions from his project team and that "continuing revision of the estimates by the people doing the work was a great help". Planning was one way of aligning the team, ensuring they were working together towards the same set of goals and milestones. In this way planning supported shared goals, which, as discussed earlier in this chapter, was one way of enhancing team cohesion.

Latent Themes

Having explored the dominant manifest themes, latent themes were investigated through interpretive analysis (R.E. Boyatzis, 1998) of the interviews. The first of the two latent themes identified relates to aspects that project managers omitted to discuss but which became apparent through broader analysis of the data. The second theme is the project managers' use of the pronoun “we” versus “they” when referring to their project team.

Project Manager Bias

There were some biases implied in the responses from the project managers. By combining the project managers’ responses and giving consideration to their likely biases and blind spots, it was possible to get a better understanding of the major factors influencing productivity. There are three ways in which project manager bias is evident in the interviews. The first is consideration of what project managers did *not* discuss. For example, the interviewees were not able to offer an objective reflection of their own competency and performance on the projects. Project management competency has been shown to be an important factor in project success (Ehsan, et al., 2010; Thite, 1999) but none of the project managers identified shortcomings in their own competency. Therefore, this is one factor which may have influenced productivity and cannot be ruled out in this study.

The second aspect of project manager bias was blind spots, where project managers do not identify critical factors as important influences of productivity when it is likely that they are critical. For example, one important factor that has been found to affect software development productivity is team size (P. Hill, 2010). The smaller the team the more productive a project generally is. Hill (2010) identified that once team size increases above five, productivity is negatively affected. In the projects studied team size ranged from 3 to 27 but no project managers discussed this as either positively or negatively impacting productivity. Similarly the software development language (for example Java, Visual Basic, Cobol) used is widely identified as one of the most significant factors affecting productivity (Boehm, et al., 2000b; P. Hill, 2010; C. Jones, 2008) but no project managers identified this as affecting productivity. This may be because they saw these factors as being outside of their control. These aspects may simply be seen as part of the context they were required to manage within and therefore not considered in their mind to be a variable that can be influenced. For these reasons, it is not possible to rule out these factors as having affected productivity in the projects studied, simply because no project managers discussed these aspects. It simply shows that project managers did not identify them as significant. However, this was compensated to some degree by gathering quantitative COCOMO II data on all of these factors, the results of which are reported in *Chapter 5 – Results and Analysis of Project data*.

The third and final area of bias was from project managers being overly optimistic in their assessment of their project. This tendency was evident with some project managers assessing their projects as productive, although the productivity metrics suggested the project was very unproductive. For example, when Jim was asked whether his project was productive, he replied “from what was delivered, it was excellent value for money”. However, the productivity achieved on that project was one of the worst in the sample.

Use of 'We' or 'They'

Some project managers used "they" or “the team” when referring to the project team and others used "we". The use of “we” suggests the speaker sees themselves belonging to the group referred to, while the use of “they” implies they see themselves as outside of the group being referred to (Auer, 2002; Gumperz, 1982; Sebba & Wootton, 1998). An analysis of the language used by project managers in response to the questions “was the project productive?” and “what affected productivity?” provides some insight as to how they saw their own role in the team’s productivity. Analysis of 14 projects’ productivity data and interviews show that project managers who referred to their teams as “we” had better productivity than those who used a mix of “they” and “the team” (see *Appendix K – Use of “We” or “They” to Refer to Software Team* for the supporting data). All project managers used “we” to refer to the team at some stage in the interview. However in five of the interviews, “we” was predominantly used in response to the two productivity questions whereas in the other six interviews, there was greater use of “they” or “the team” . The use of “we” rather than “they” or “the team” could suggest differing levels of perceived responsibility on the part of the project manager for the project outcomes. The link between “we” and better performing projects implies that if the projects performed badly, and the project manager refers to the project team as "they", then the project manager may see the bad performance due to the project team and not themselves. The use of "we" or "they" by project managers may also indicate how a project manager views their role. For example one who uses "they" may have a more dictatorial leadership style (Bruce & Langdon, 2000; Sukhoo, et al., 2005) while one who uses "we" may be more participative or consultative (Bass, et al., 1975).

Summary of Results and Analysis of the Interviews

Three of the four major factors that project managers identified as affecting productivity were influenced by aspects of ethnic diversity and the most significant productivity factors were also all interconnected. Ethnic diversity within software teams was found to have a negative impact on the quality of communication within the team and this is significant given that communication was the most important factor affecting software development productivity. The ethnic diversity had positive effects on team cohesion (in some projects) and improved team relationships. Given that team cohesion was the second most significant factor affecting productivity and ethnic diversity affected team cohesion, this shows the potential for ethnic diversity to influence productivity through its effect on team cohesion. In addition, the positive influence of diverse perspectives on team capability is important given that team capability was identified by project managers as an important productivity factor. Underpinning the influence of diversity in productivity were trust and team relationships. Trust was identified by project managers as a critical enabler for effective communications within the project team as well as being important for achieving team cohesion. Furthermore the ability to improve relationships between team members was enhanced where an environment of trust was created by the project manager.

A number of the major productivity factors identified in the interview responses are interrelated. Communications are both a cause and an outcome. Team cohesion is impacted by the quality of communication and communication is improved when teams are cohesive. Agile practices improved both communication and team cohesion. The agile practices were identified as directly improving productivity as well as indirectly affecting productivity through improved communication and team cohesion. Requirements quality is enhanced by good communication and team cohesion. This interrelatedness makes it difficult to identify the single driver for productivity, but instead implies these factors work together to improve productivity. The interrelationships between the productivity factors are considered important and provide insight into the influence of ethnic diversity on software development productivity.

This analysis of the most significant productivity related factors and the diversity-related findings serve to amplify the importance of the influence of ethnic diversity on software development productivity. It is only in the context of the significant productivity factors that the relevance of ethnic diversity can be fully understood. The next chapter continues to examine the influence of team diversity by reporting on the findings from analysis of the quantitative project data. Finally the qualitative and quantitative results are brought together to provide the overall findings regarding the influence of ethnic team diversity on software development productivity.

CHAPTER 5 – RESULTS AND ANALYSIS OF PROJECT DATA

Introduction

This chapter presents the analysis of ethnic diversity and productivity based on the quantitative data. The structure of this chapter is based on the research questions stated in *Chapter 1 – Introduction*. Before considering the research questions using the quantitative data, the chapter begins by presenting the descriptive statistics for the data which combines COCOMO II productivity variables with ethnic diversity indices. This provides an overview of the data and confirms its suitability for the analysis methods employed. The second section reports on the factors correlated with productivity as well as the correlations between ethnic diversity and productivity. Examining these associations provides information about how ethnic diversity was associated with the productivity of software development teams.

The third section reports on the aspect of organisational context that had most impact on productivity and presents how that altered correlations between ethnic diversity and productivity. As government projects had significantly different productivity to non-government projects, the correlations between ethnic diversity and other variables have been analysed separately within these two groups of projects. Finally a summary brings together the results and findings from the analysis of the project data.

Descriptive Statistics

Descriptive statistics provide a useful overview of the data (Boudreau, et al., 2001; Gable, 1994; Onwuegbuzie, Slate, Leech, & Collins, 2007) and are presented in the following three tables. An explanation of each of the variables can be found in *Appendix F – List of Variables*. The percentages in Table 11 do not add to exactly 100% in all cases due to rounding.

Table 11. Data frequencies for categorical variables of projects

Variable	Projects	Frequencies			
Project manager gender	19	Female 21%	Male 79%		
Project manager ethnicity	19	Asian 16%	European 84%		
Project manager age	19	30-39 11%	40-49 37%	50-59 42%	60-69 11%
Project success	17	Major problems 18%	Some problems 6%	Successful 53%	Very successful 24%
Enhancement or New Development	19	Enhancement 68%	New Development 32%		
Main programming language	16	Java 50%	.NET 31%	Other 19%	
Use of agile practices	15	None 53%	Some 40%	Extensive 7%	
Primarily a software producer	19	Yes 42%	No 58%		
Government organisation	19	Yes 37%	No 63%		

For most variables, data was captured for all 19 projects. However, in some cases where it was not possible to interview the project manager, some items of data could not be collected. The 'projects' column on Table 11 lists how many projects on which data for each variable was collected. The first two rows of Table 11 show that the sample had a weighting towards male project managers and also project managers who identified their ethnicity as European, or a sub-group of European including New Zealand European, English and Scottish. Table 11 also shows that most project managers were in the age group 40 – 59. Most projects were described as successful or very successful by the project manager, although four were described as either having some problems or major problems. Java was the main programming language used and most projects involved making enhancements to existing software rather than developing new software. Only one project made extensive use of agile practices, with six using some agile practices and eight not using any.

Table 12 presents the data frequencies for each of the Likert type variables captured. These variables are part of the COCOMO II model and have been gathered using the rules and guidelines associated with that model (Boehm, et al., 2000a). The percentages in Table 12 do not add to exactly 100% in all cases due to rounding.

Table 12. Data frequencies for Likert type variables

Variable	Valid	Very Low	Low	Nominal	High	Very High	Extra High
Analyst capability	15	0%	13%	13%	60%	13%	0%
Language and tool experience	15	0%	7%	13%	40%	40%	0%
Personnel continuity	15	7%	20%	7%	13%	53%	0%
Precedentedness	17	12%	18%	12%	18%	24%	18%
Development flexibility	17	24%	35%	29%	6%	6%	0%
Risk resolution	17	12%	18%	0%	41%	29%	0%
Team cohesion	17	6%	12%	18%	35%	29%	0%
Required software reliability	17	0%	29%	18%	53%	0%	0%
Developed for reuse	17	0%	24%	12%	6%	18%	41%
Documentation required	17	0%	29%	59%	12%	0%	0%
Software complexity	15	0%	27%	13%	53%	7%	0%
Platform volatility	15	7%	60%	27%	7%	0%	7%
Programmer capability	15	0%	0%	27%	40%	33%	0%
Applications experience	15	13%	7%	33%	27%	13%	0%
Platform experience	15	7%	0%	20%	60%	13%	0%
Use of software tools	15	0%	40%	33%	13%	13%	0%
Multisite development	15	0%	13%	7%	20%	40%	20%
Required development schedule	15	0%	13%	87%	0%	0%	0%
Process maturity	15	13%	20%	47%	20%	0%	0%
Requirements volatility	15	0%	33%	13%	47%	7%	0%

Table 12 shows that the projects in the sample generally represent a broad range of values for the productivity factors. For most variables there is at least one project for each of the possible rating values. For some variables there is an obvious skew, such as for programmer capability. This shows that all project managers believed their programmers to be of at least nominal ability (that is, at least average) and most thought they were either highly capable (40%) or very highly capable (33%). A rating of high for programmer capability indicates the project manager has assessed the programmers as being in the top 25% of all programmers and very high is the top 10% of programmers. It should be noted that these factors are all assessed by the project managers, and therefore represents their views. This is the recommended way to capture the COCOMO II productivity factors (Boehm, et al., 2000b) and has been found to provide a relatively accurate assessment of them (Chulani, et al., 1999) as project managers are generally the best people to assess the characteristics of their team (Ehsan, et al., 2010; Wang, 2009).

Table 13 presents the descriptive statistics for the interval data captured. A list of all variables with a full description is presented in *Appendix F – List of Variables*.

Table 13. Descriptive statistics for interval data

Variable	N	Mean	Std. Deviation	Min.	Median	Max.	Spread
Years spent managing projects	19	15.74	5.45	6	18.00	25	19
Project length in months	19	10.53	6.594	4	9.00	24	20
Normalised project hours (total hours of effort)	16	5,487	7,743	117	899	21,331	21,214
Unadjusted function points (software size)	16	242	393	28	105	1631	1,603
Normalised project delivery rate	16	19.44	17.29	1.60	11.19	54.15	52.55
Team size	15	9.80	7.28	3	9.00	27	24
Percentage of females	15	27.40	16.38	0	25.00	64	64
Number of ethnicities in the team	15	2.73	.80	2	3.00	4	2
Blau's ethnic diversity index	15	.55	.19	.00	.56	.81	.81

Table 13 shows that the average productivity (normalised project delivery rate or NPDR) for the sample is similar to that of the software industry internationally, based on data from the International Software Benchmarking Standards Group (ISBSG). The mean NPDR for the 3404 ISBSG projects with productivity data is 19.2 hours per function point and the median is 10.7 (International Software Benchmarking Standards Group, 2009). This compares with a mean of 21.48 and a median of 11.19 for the sample in this study. This average for the 16 projects for which productivity data was available is similar to the international average for software development productivity. Blau's ethnic diversity index for the projects ranged from 0 to .81 with only one project being below 0.25. This is significant because any value above .25 is considered to have a relatively high degree of diversity (Richard, et al., 2004) indicating that most of the teams were ethnically diverse.

In summary, the descriptive statistics information presented provides an overview of the data. The next section presents the results of correlation analysis, showing the ethnic diversity variables correlated with software development productivity.

Correlation Analysis

Correlation analysis was used to identify associations that were significant using an alpha level of .05 in order to ensure that only significant relationships were identified and investigated further. An alpha level of .05 has been used traditionally as a standard (Hubbard & Lindsay, 2008) in the social sciences (Gaur & Gaur, 2009) and in studies of software development (Q. Liu & Mintram, 2005; Lokan, 1999; Nguyen, Steece, & Boehm, 2008). Although the origins of .05 as the threshold for statistical significance are rather arbitrary (Cohen, 1994), it does represent a level of likelihood that people can generally accept (Stigler, 2008).

Projects with missing values for one or both of a pair of variables are excluded from the analysis. As each coefficient is based on all the projects that have data on that particular pair of variables, the maximum information available is used in every calculation. A consequence of this is that the correlations reported are based on a varying number of projects. Therefore, for each pair of variables, the number of projects used is reported (N).

For the interval and ordinal data captured, no transformation of the raw data is required in order to undertake ranked correlation analysis. The variables included from COCOMO II are Likert type variables which are commonly treated as ordinal (Allen & Seaman, 2007; Brown, 2011; Clason, et al., 1994; Jamieson, 2004) and are suitable for ranked correlation analysis (Jakobsson, 2004;

Sheskin, 2007). For dichotomous or binary data it was necessary to assign an arbitrary number to each value (sometimes called a dummy variable (Gujarati, 1970; LaVeist, 1994)). This enabled correlation analysis with the other variables which include both ordinal and interval data. This has allowed significant associations with these binary variables to be identified by interpreting the results based on the numbers assigned to each variable. The values 0 and 1 have been used as these are the values typically assigned to dichotomous variables (Norušis, 2008). There were five dichotomous variables which are listed in Table 14 along with the number assigned to each value.

Table 14. Values assigned to dichotomous variables

Variable	**Value**	**Meaning**
Is the organisation a government organisation	0	Non-government
	1	Government
Is software production the primary activity	0	No
	1	Yes
Project manager gender	0	Male
	1	Female
Project manager ethnicity (only two were reported)	0	European
	1	Asian
Enhancement or New Development (TYPE)	0	Enhancement
	1	New development

Using ranked order correlation analysis where one variable is dichotomous, is similar to performing a two sample t-test. However with over 30 variables being examined for correlations under all logical pairing, there would be many hundreds of tests to perform. Therefore it was more efficient to perform ranked correlation analysis across all variable pairs in one operation. Where correlation analysis shows significant associations with binary variables, the strength of these associations can be investigated further using t-tests (Coolican, 2009; Green & Salkind, 2011).

Therefore, for correlations between a dichotomous variable and an interval (or continuous) variable, a two sample t-test was used to further investigate the relationship. Although the data has not been proven to be normally distributed, the results of two sample t-tests have been found to be relatively resilient to deviations from normality (Boneau, 1960; Cohen, 1988; Edgell & Noon, 1984).

While the use of correlation provides no statistical evidence of causal relationships, some explanations are discussed and considered in the analysis of the data. In some cases these explanations are based on previous research which has demonstrated a causal relationship. In other cases, the explanations are based on other additional information available, such as that gathered in the project documentation and interviews. This approach reflects a departure from a purely quantitative approach (Feuer, et al., 2002; Henson, et al., 2010) and instead provides a more interpretive, context-aware understanding (Harrits, 2011; J. A. Maxwell, 2004) of the influence of ethnic diversity on software development productivity.

Ethnic Diversity and Productivity

To help answer the research questions, this section examines correlations between the ethnic diversity variables for the project teams and the productivity variables. As discussed in *Chapter 3 – Method*, two measures of ethnic diversity were used for the software project teams analysed. The first measure used was Blau's Ethnic Diversity Index which had no significant correlations to other variables. The second measure of ethnic diversity used is the number of ethnicities in the team. This measure is not the best indicator of ethnic diversity as the larger the team, the larger the number of ethnicities likely to be in the team. However it does represent an important aspect of team diversity as it provides a measure of the variety of ethnic groups present in the team. Analysis showed a number of significant correlations for the number of ethnicities in the team, including the variables correlated with productivity as well as with productivity itself.

Figure 5 shows all variables correlated with the number of ethnicities and indicates whether each pair of variables is positively or negatively correlated. As productivity is a ratio of inputs (hours) to outputs (software), a high value indicates it took more hours to produce each unit of software. Therefore a high productivity value indicates poor productivity and as a result, all of the factors positively correlated with productivity were associated with poor productivity. For example team size is positively correlated to productivity and therefore the larger the team, the higher the productivity figure, indicating that larger teams took longer to produce each unit of software.

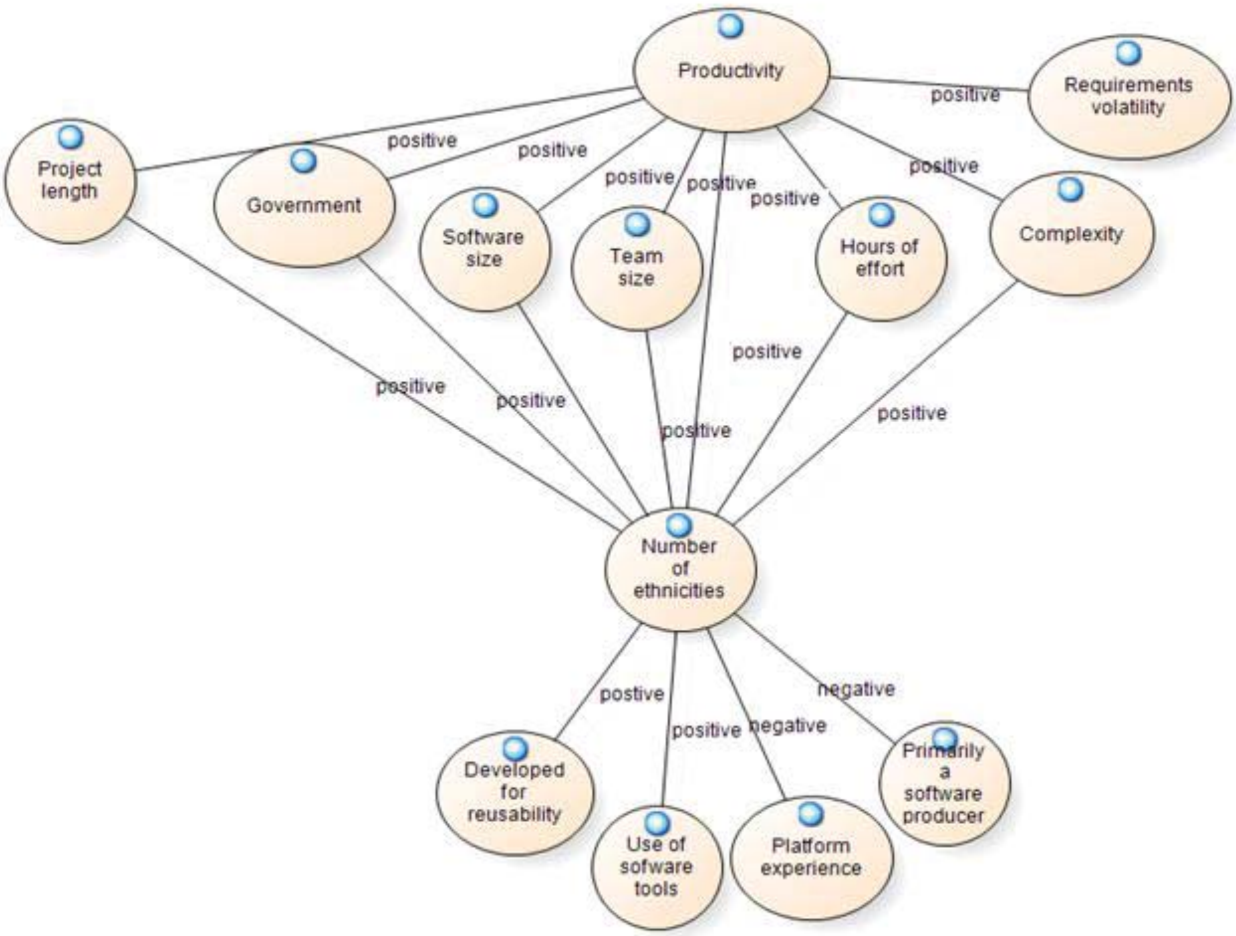


Figure 5. Variables correlated with the number of ethnicities in the team

The following sections report the degree of correlation and the significance for each of the 18 correlations shown in Figure 5. This includes all variables correlated with the number of ethnicities and productivity. Matrices showing all correlations are presented in *Appendix L – Kendall Rank-Order Correlations*.

Correlations between Ethnic Diversity and Productivity

The first correlation shows a positive association between the number of ethnicities in the team and productivity, τ=0.685, *n*=14, *p*=0.002. This indicates that teams with a greater variety of ethnicities had worse productivity as the higher the number of ethnicities, the more hours of effort required to produce each unit of software. The reasons for this association are explained by analysing the correlations between the number of ethnicities and the factors that have previously been found to influence productivity on software projects. Therefore, the following sections examine these correlations and explain why this association exists between poor productivity and ethnic diversity.

Correlations between Ethnic Diversity and Productivity Factors

Ethnic diversity, measured by the number of ethnicities, was correlated with six variables that were correlated with productivity. These six productivity factors are team size, project length, total hours of effort, product size, product complexity and whether the project was undertaken by government.

Project size, ethnic diversity and productivity

Table 15. Ethnicity and productivity variables correlated with productivity factors

No.	Variable 1	Variable 2	N	τ	Sig.
1.	Total hours of effort	Productivity	16	.550**	.003
2.	Project length in months	Productivity	16	.593**	.002
3.	Team size	Productivity	14	.709**	.001
4.	Software size in function points	Productivity	16	.377*	.043
5.	Number of ethnicities	Total hours of effort	14	.658**	.003
6.	Number of ethnicities	Project length in months	15	.764**	.000
7.	Number of ethnicities	Team size	15	.683**	.002
8.	Number of ethnicities	Software size in function points	14	.523**	.019

*p<.05 **p<.01

The first row in Table 15, shows a positive correlation between the total hours of effort spent on the project and the hours it took to produce each unit of software (productivity). Put another way, the larger the project (measured in total hours of effort) the less efficient the software development team were. The second correlation also suggests that large projects were unproductive with a positive correlation between the length of the project and productivity. The longer the project, the more hours it took to produce each unit of software. The next correlation further supports the association between large projects and poor productivity as team size was positively correlated with productivity. The more people there were in the team, the longer it took to produce each unit of software. Finally, the fourth correlation shows that size of the software being developed was positively correlated with productivity, indicating that the larger the software product being developed, the longer it took to produce each unit of software. This correlation also supports the association between large software projects and poor productivity. Therefore, the correlation between productivity and the first four productivity factors (rows 1 - 4 in Table 15) show that larger software projects were less productive than smaller ones. Projects that were larger when measured using total hours of effort, length in months, the size of the team and the product size in function points, had worse productivity.

The second half of Table 15 (rows 5 - 8) shows the significant correlations between the number of ethnicities in the team and the same four measures of project size. The first of these (row 5) shows a positive correlation between the total hours of effort spent on the project and the number of ethnicities in the team. Put another way, the larger the project was (measured in total hours of effort), the more ethnicities there were in the team. The next row shows a positive correlation between project length and the number of ethnicities. This indicates that the longer the project, the more ethnicities there were in the team. This supports row 5 in showing that larger projects had more ethnicities. The seventh row shows that team size was positively correlated with the number of ethnicities, indicating that larger teams had more ethnicities. Finally the number of ethnicities was positively correlated with the size of the software being developed. Therefore, projects where the software being developed was large were more likely to have more ethnicities. Overall, projects that were large, using four different measures were more likely to have more ethnicities.

The correlations between poor productivity, more ethnicities and large projects is most likely to be because a larger team is more likely to include a broader range of people, including different ethnicities, and large teams are associated with poor productivity (Fried, 1991; P. Hill, 2010; K. Maxwell, et al., 1996). Previous studies into the effect of project size on software development productivity have found that large projects can be less productive than small ones (Boehm, 1981; Shepperd, 2007). This tendency for large software development projects to be less productive than small ones has been referred to as “diseconomies of scale” (Boehm, 1981, p. 190). Whether diseconomies or economies of scale exist on software projects has been widely debated (Banker, Chang, & Kemerer, 1994). It has been argued that this may depend on other factors, such as process maturity (Boehm, et al., 2000b). Another perspective is that there may be a non-linear relationship between size and productivity (Kitchenham, 2002). For example, on smaller projects there may be economies of scale but for larger projects diseconomies of scale occur. Despite these various theories, it is still generally accepted that larger software projects are less productive than smaller ones (Shepperd, 2007). This view is consistent with the correlations observed in the software projects analysed in this study.

The cause for diseconomies of scale on software projects is generally attributed to the fact that software development is a communication intensive activity and the more team members there are, the more communication paths exist (Boehm, 1981; Brooks, 1995; Ganssle, 2008). This creates the potential need for exponentially more interactions between project team members as projects become larger (Ganssle, 2008). While having a variety of ethnicities has been found to have both positive and negative effects, the negative impacts often relate to communication challenges (Egan, et al., 2006; Shachaf, 2008). Some negative impacts of communication related to ethnic diversity were also reported in *Chapter 4 – Results and Analysis of Interviews*. The impact of ethnic diversity on communication was evident in the projects and this accounts, at least in part, for the association between ethnic variety in the team, large projects and poor productivity.

Product complexity, ethnic diversity and productivity

Table 16. Ethnicity and productivity variables correlated with complexity

No.	**Variable 1**	**Variable 2**	**N**	τ	**Sig.**
1.	Number of ethnicities	Complexity	15	.766**	.001
2.	Complexity	Productivity	14	.674**	.002

*p<.05 **p<.01

The fifth productivity factor correlated with the number of ethnicities (row 1 of Table 16) indicates that projects which had teams with more ethnicities were more technically complex. The correlation between productivity and complexity (row 2) shows that more technically complex projects were less productive. It logically follows that more complex tasks take more effort to complete and this relationship has been observed in software development projects in previous research (Boehm, et al., 2000a; Heijstek & Chaudron, 2009; K. Maxwell & Forselius, 2000). The reason for ethnic diversity being associated with complex projects is less clear, but complex projects would benefit from improved innovation and problem solving capability associated with having team members with diverse perspectives.

Government projects, ethnic diversity and productivity

Table 17. Ethnicity and productivity variables correlated with government

No.	Variable 1	Variable 2	N	τ	Sig.
1.	Number of ethnicities	Government organisation	15	.698**	.006
2.	Government organisation	Productivity	16	.542**	.013

*p<.05 **p<.01

The sixth and final variable with a significant correlation to the number of ethnicities (row 1) and to productivity (row 2) was whether a project was being undertaken by a government organisation. The first of these correlations shows that the number of ethnicities was positively correlated with whether a project was being undertaken by a government organisation. This indicates that software project teams in government organisations were also more likely to contain more ethnicities. Equal employment opportunity programmes are mandated and also widely communicated within government organisations in New Zealand. While it is illegal to discriminate based on ethnicity in New Zealand according to the 1993 Human Rights Act (New Zealand Parliament, 1993), the state sector takes this further by requiring government departments to have equal employment programmes (New Zealand Parliament, 1988) and work towards diversity across the government's work force (State Services Commission of New Zealand, 2008).

The second of these correlations shows that productivity was positively correlated with whether a project was being undertaken by a government organisation. This correlation indicates that on government projects it took more hours of effort to deliver each unit of software than on non-government projects. The likely reasons for government projects being significantly less productive than non-government projects are discussed in the sub-section *The Impact of Being a Government Organisation* later in this chapter.

Ethnic Diversity with Factors Previously Found to Affect Productivity

In order to further examine the influence of ethnic diversity in software development teams, correlations between ethnically diverse teams and factors previously found to affect software development productivity have been examined. Although these factors were not correlated with productivity in this study, they are correlated with ethnic diversity and the fact that they have previously been found to affect productivity means they provide insight into the ways ethnic diversity influences productivity. These correlations are shown in Table 18.

Table 18. Ethnicity variables correlated with factors previously affecting productivity

No.	**Variable 1**	**Variable 2**	**N**	τ	**Sig.**
1.	Number of ethnicities	Developed for Reusability	15	.618**	.007
2.	Number of ethnicities	Platform Experience	15	-.559*	.020
3.	Number of ethnicities	Use of Software Tools	15	.497*	.035
4.	Number of ethnicities	Software producer	15	-.573*	.024

*p<.05 **p<.01

The first of these correlations (row 1) shows that the number of ethnicities was positively correlated with software being developed for reuse. This indicates that teams with more ethnicities were more likely to be working on software projects where there was a requirement to create software that was reusable in other systems. According to the research undertaken on 161 software projects to develop the COCOMO II model, projects where software was developed for reuse required greater effort and were therefore less productive. This finding is also supported by other empirical research into the effect of developing for reuse (Addy, Mili, & Yacoub, 1999; C. Jones, 2008; Orrego, Menzies, & El-Rawas, 2009). Therefore, the results shown in row 1 provide some further explanation of why ethnic diversity was associated with poor productivity, as ethnically diverse software teams were more likely to be working on projects where the requirement for reuse was higher.

Projects with more ethnic groups had less experience with the technical platform used for those projects. The number of ethnicities is negatively correlated with platform experience (row 2 in Table 18) so teams with a high number of ethnicities had a low level of platform experience. According to COCOMO II, less experience with the platform leads to poor productivity, so this also helps to explain why high ethnicity diversity was associated with poor productivity. More specifically, the results shown in rows 1 and 2 of Table 18 suggest that it was not ethnic diversity that accounted for poor productivity, but simply the fact that the ethnically diverse teams in this study had to develop for reuse and had low platform experience, and these factors that are more likely to have negatively affected productivity.

The number of ethnicities was positively correlated with the use of software tools (row 3 in Table 18). Projects with more ethnic groups were more likely to make greater use of software tools to support their software development and according to COCOMO II, greater use of tools should make projects more productive. As discussed in *Chapter 4 – Results and Analysis of Interviews* a number of the tools used on the software projects studied involved computer mediated communication. The use of computer mediated communication tools is one way of addressing communication challenges that can exist within ethnically diverse software teams (Nam, et al., 2009; Shachaf, 2008). Therefore, the use of software development tools that assist with communication were greater in ethnically diverse teams and this is likely to have improved productivity.

Finally, row 4 in Table 18 shows that the number of ethnicities was negatively correlated with whether the organisation undertaking the project was primarily a software producer. Organisations which were primarily software producers had fewer ethnic groups in their teams. While there is no conclusive research on whether organisations that are primarily software producers are more productive than other software producers, they are often considered to be more productive and this is one justification used for the outsourcing of software development (Dibbern, et al., 2004). This correlation adds further support to the conclusion that it was not ethnic diversity that accounted for poor productivity, but simply the fact that the ethnically diverse teams in this study were more likely to be in organisations that are typically less productive.

Therefore, from this analysis of the associations between ethnic diversity and productivity factors, ethnic diversity is not directly attributed to influencing software development productivity.

Requirements Volatility and Productivity

One further factor that was positively correlated with productivity was requirements volatility, τ=0.436, n=14, p=0.048. This indicates teams where the requirements changed significantly and frequently had worse productivity as the higher the level of requirements volatility, the more hours of effort were required to produce each unit of software. Requirements volatility was not correlated with the number of ethnicities in the team, but as most of the teams analysed had a high level of ethnic diversity, this can be considered a factor which may influence productivity in ethnically diverse software teams. That is, in the software project teams studied, which were predominantly ethnically diverse, if the requirements changed significantly and often during the project, then this negatively impacted productivity.

Factors That Altered the Influence of Ethnic Diversity

Having discussed how ethnic diversity was associated with software development productivity, this section considers what factors may have altered the influence of ethnic diversity and seeks to address the third research question. Given that the sample is relatively small for statistical analysis with 19 projects, certain statistical analysis methods which could assist in identifying the effect of other variables on the influence of ethnic diversity are not suitable. For example, as a general heuristic, multiple linear regression analysis requires at least 10 cases per independent variable (Chin, Marcolin, & Newsted, 2003; Marcoulides & Saunders, 2006). However, where the factor is a dichotomous variable, as is the case with sector (that is, government versus non-government), this allows a comparison of the two groups as the number of projects in each group is sufficient for undertaking t-tests and correlation analyses (Hill & Lewicki, 2006; Sheskin, 2007; Siegel & Castellan, 1988).

The Impact of Being a Government Organisation

The most significant factor correlated with productivity was whether or not the organisation undertaking the project was government. As organisational context is an important aspect affecting the influence of team diversity in organisations (Jehn & Bezrukova, 2004; Kochan, et al., 2003; K. Y.

Williams & O'Reilly, 1998), this has been further analysed using the quantitative data. By examining the influence of ethnic diversity in government and non-government separately, the different diversity-related relationships can be compared and contrasted.

Of the 19 projects analysed, it was not possible to collect productivity data from three of the projects and therefore they could not be included in this analysis of the correlation between government organisations and productivity. This leaves 16 projects from five organisations. The productivity achieved on these projects, measured using the normalised project delivery rate (NPDR), is shown in Figure 6 (fewer hours per function point indicates a more productive project).

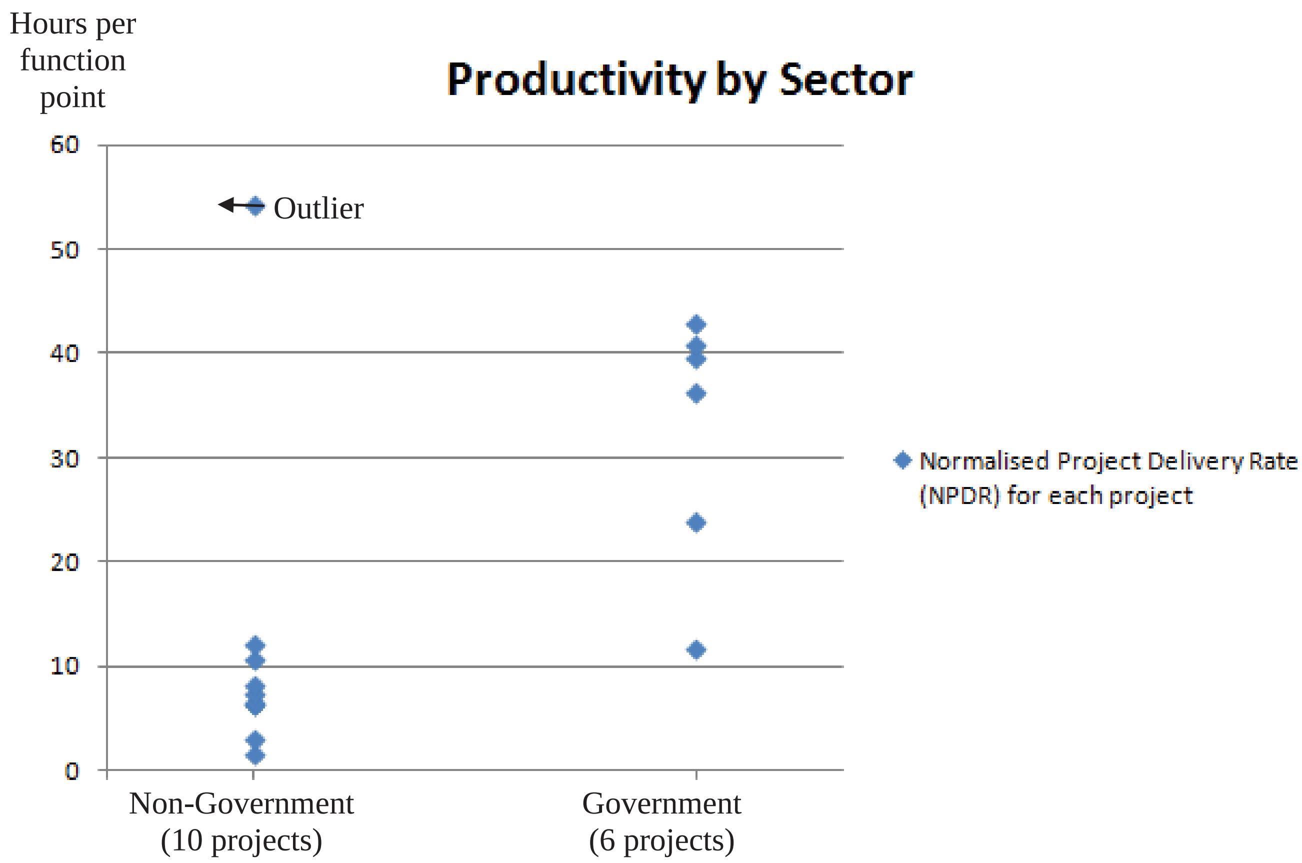


Figure 6: Productivity for government and non-government projects

Non-government software producers achieved significantly better productivity on their software projects (Figure 6). Their projects ranged from 2 to 54 hours per function point but if the outlier of 54 is excluded the range of the remaining nine non-government projects is 2 to 12. Government projects achieved a range of 12 to 43 hours per function point. Non-government projects had a lower mean productivity rate (M = 11.6, SD = 15.27) than government projects (M = 32.50, SD = 12.23). Performing a t-test on this data shows that the difference between mean productivity for each group (difference = 20.90, 95% CI: 5.11 to 36.69) was significant, $t(14) = 2.84$, $p=.013$ (two-tailed), $d = 1.21$.

The outlier which recorded the worst productivity of all the projects studied (54 hours per function point) was the largest project and had a number of complexities not present in the other projects. The project included five sub-teams, most of which were located at different physical locations. It was also a multi-platform development delivering to both mobile devices and conventional computers. Due to these significant productivity inhibiting characteristics unique to this project, it was excluded from the t-test comparison between the government and non-government projects.

Given the significantly worse productivity for government projects, further analysis has been undertaken to examine what factors are correlated to government projects. This provides additional information about the organisational context that differentiates government projects from non-government. Table 19 shows the variables that were correlated with whether or not the project was undertaken by a government organisation but excludes government correlations already discussed in this chapter (productivity and the number of ethnicities).

Table 19. Significant correlations between variables and government

No.	**Variable 1**	**Variable 2**	**N**	τ	**Sig.**
1.	Government organisation	Project length in months	19	.724**	.000
2.	Government organisation	Total hours of effort	16	.728**	.001
3.	Government organisation	New development	19	.655**	.002
4.	Government organisation	Software producer	19	-.651**	.003
5.	Government organisation	Developed for Reusability	17	.638**	.006
6.	Government organisation	Platform Experience	15	-.667**	.007
7.	Government organisation	Use of Software Tools	15	.636*	.011
8.	Government organisation	Multisite development	15	.628*	.012
9.	Government organisation	Team size	15	.624*	.013
10.	Government organisation	Software size in function points	16	.589*	.016
11.	Government organisation	Complexity	15	.594*	.020

*p<.05 **p<.01

Eleven variables were correlated with whether the organisation undertaking the software project was a government department. The first two listed in Table 19 show that government projects were positively correlated with both total project hours and project length meaning government projects were larger and longer. This provides some explanation of why government projects were less productive as these factors have previously been found to negatively impact productivity. Correlations 9 and 10 also indicate that government projects were larger when measured in terms of team size and software size.

Government projects were generally new developments rather than enhancements to existing systems and industry benchmark data shows that new developments have better productivity than enhancements (International Software Benchmarking Standards Group, 2009; C. Jones, 2008). Therefore if government organisations undertook more enhancement projects than non-government, this could help explain the difference in productivity between government and non-government.

However, the correlation between government organisations and new developments shown in row 3 of Table 19 indicates the opposite is true in the projects studied. Projects undertaken by government were more likely to be new developments and therefore, based on this factor, should have been more productive than those done by non-government. Therefore, it is unlikely that the *type* of software projects (that is enhancement or new development) led to government projects being less productive. The fourth correlation simply shows that government agencies are not primarily software producers which is unsurprising as no government agencies exist primarily to produce software.

Out of all eleven factors associated with the sector in Table 19, only two have been found to improve productivity. One is new developments (discussed above) and the other is the seventh correlation which shows that government projects made greater use of software tools. Most of the factors associated with government projects have a negative effect on productivity according to COCOMO II (Boehm, et al., 2000b). This included developing software for reuse, less platform experience, higher software complexity and undertaking software development across multiples sites.

Government project teams had a greater number of ethnic groups. This can require a greater focus on communication and co-ordination in order to ensure the benefits of ethnic diversity occur such as in problem solving and innovation (Brandes, et al., 2009; P. Richardson, 2005; Winkler & Bouncken, 2009). As government teams were also larger and the projects were longer, this increases the need for focusing on communication and coordination. The greater use of software tools on government projects was one way to help address this challenge as the tools used on software projects often seek to improve communication and coordination across the team and across time (Candrlic, et al., 2006; Cook & Churcher, 2006; de Souza, et al., 2007; Tiwana, 2008). These tools deal with the different aspects of communication and coordination which are critical for effective software projects, especially large projects and projects where there is potential for misunderstandings.

In summary, most of the factors associated with government projects are likely to have negatively impacted productivity. These factors are that government projects were larger and longer than non-government projects, members had less platform experience, they were focussed on developing software for reuse and undertook development across multiple sites. Potentially counteracting this was that government projects were mostly new developments, something that is generally associated with improved software development productivity. There was greater use of software tools on government projects which has been found to improve productivity and may also support co-ordination and communication across the larger and more ethnically diverse government software projects.

Recent research into software development projects in government has also shown government projects to be less productive than non-government (Congalton, 2011). This analysis of the 5000 international software projects in the International Software Benchmarking Standards Group (ISBSG) database found that other factors accounted for this productivity difference. A high proportion of projects undertaken by government were multi-platform and projects done by government had the lowest proportion of mainframe projects. When compared to non-government projects, the proportion of standalone projects done by government was considerably lower. Projects undertaken for government (that is, government projects that were outsourced) were more likely to be mainframe, suggesting that mainframe development for government was generally outsourced. Projects completed for government organisations were also larger in size than non-government projects with the outsourced government projects being, on average, the largest overall. The sample of 19 New Zealand projects used in this study has some characteristics consistent with the ISBSG data. For example, software projects undertaken by government were larger when measured using function points or hours of effort. However the difference between government and non-government productivity is more pronounced in the New Zealand sample. Given that both ISBSG and the New Zealand projects show that software projects undertaken by government agencies had worse productvity, this relationship is considered signficant. In summary, the ISBSG data indicated that government projects had worse productivity and were larger when measured using function points or hours of effort with no clear reason why. In the New Zealand sample used in this study, the government also had worse productivity, and were larger and longer.

Comparison of the Influence of Ethnic Diversity in Different Sectors

As the productivity and other characteristics varied significantly between government and non-government projects, ethnic diversity variable correlations have been examined within each sector separately. This was to obtain a more accurate understanding of the influence of ethnic diversity in the 19 software projects studied and how the sector may impact the influence of ethnic diversity on productivity

The number of projects in each sector is small (seven government and twelve non-government projects). However, ranked correlation analysis is a technique ideally suited to small samples (Hill & Lewicki, 2006; Siegel & Castellan, 1988) and has been used effectively with samples as small as five (Sheskin, 2007). While this sample is too small to be representative of all software projects, it assists in analysing the associations within the sample and understanding if the organisational context impacted the influence of ethnic diversity.

In the sample of seven government software projects analysed there were no significant correlations with either of the ethnic diversity variables (the number of ethnicities or the ethnic diversity index) and there were no variables correlated with productivity. In the sample of 12 non-government software projects analysed, there were three significant correlations with the ethnic diversity index (Table 20) but no variables correlated with productivity.

Table 20. Significant correlations for non-government projects

No.	Variable 1	Variable 2	N	τ	Sig.
1.	Number of ethnicities	Multisite development	8	.889**	.003
2.	Number of ethnicities	Use of Software Tools	8	-.900**	.002
3.	Number of ethnicities	Complexity	8	.729*	.040

*p<.05 **p<.01

The first correlation shows that projects with ethnically diverse teams were more likely to be undertaken across multiple sites. Projects where the team is located across multiple sites have been found to be less productive as more effort is expended in maintaining effective communication and co-ordination (Boehm, et al., 2000b; Daim, et al., 2012; Herbsleb, 2007). There is also greater possibility of misunderstandings as face-to-face communication is generally considered most effective (Burgoon, et al., 1994; Modaff, et al., 2008). Ethnic diversity can also give rise to similar issues where there are communication challenges and the potential for misunderstandings (Brandes, et al., 2009; P. Richardson, 2005; Winkler & Bouncken, 2009). With both factors having the potential to inhibit effective communication, the negative effects of ethnic diversity may be more likely to occur on multi-site projects and the benefits of ethnic diversity less likely to arise. Secondly, the number of different ethnicities was negatively correlated with the use of software tools. This shows that ethnically diverse teams on non-government projects made less use of software tools and such tools can assist communication within software projects. Thirdly, the ethnically diverse teams were more likely to be working on complex software projects.

In summary, all three productivity factors associated with greater ethnic diversity in the non-government projects have previously been found to negatively impact productivity (multisite development, less use of software tools and greater complexity). Furthermore, these factors would generally be expected to inhibit communication and could therefore be expected to compound any communication challenges. Despite this, non-government projects were significantly more productive than the government projects.

Summary of Results and Analysis of Project Data

Productivity was positively correlated with the number of ethnic groups in the team showing that teams with more ethnicities had worse productivity. The number of ethnic groups was also correlated with six of the seven factors that were correlated with productivity. These six productivity factors are team size, project length, total hours of effort, product size, product complexity and whether the project was undertaken by government. It is unlikely that having multiple ethnicities caused these factors to occur as all of these variables correlated with ethnic diversity have been previously shown to negatively impact productivity. Therefore, rather than multiple ethnicities causing poor productivity, it is more likely that projects which are larger, more complex or undertaken by government tended to have more ethnicities in their teams. Four other factors which have previously been associated with productivity were also correlated with the number of ethnicities in each team. Teams with more ethnicities were likely to be creating software components which were reusable but had less experience with the platform being used. These are both factors which have been found to inhibit productivity. However, across all projects, teams with more ethnic groups made greater use of software tools which has been found to improve productivity and can help to enhance communication within ethnically diverse software teams. Finally, organisations which were primarily software producers had fewer ethnic groups in their project teams.

Government projects were significantly less productive than non-government projects and this difference represented a cluster of correlations. Most of the factors associated with government projects are likely to have negatively affected productivity. These factors are that government projects were larger and longer than non-government projects, they had less platform experience, focussed on developing software for reuse and undertook development across multiple sites. These are all factors which have previously been found to negatively impact software development productivity. Within the non-government projects, ethnically diverse teams were associated with multisite development, less use of software tools and greater complexity, all of which have been found to negatively affect productivity. However, despite this non-government projects were significantly more productive.

These findings based on a quantitative analysis of the project data provide some insights into what factors may have affected productivity and what influence ethnic diversity may have played in the productivity of the software development projects studies. The following chapter brings together these quantitative findings with the results and analysis of the interviews to identify important themes and present answers to the research questions.

CHAPTER 6 – SYNTHESIS AND CONCLUSIONS

The purpose of this mixed methods research was to investigate how ethnic diversity in software development teams influences the productivity those teams achieve. Nineteen software projects undertaken in New Zealand were analysed by interviewing project managers as key informants. Project documents were also used to gather productivity data. Fourteen of the projects had ethnically diverse teams and these were used as the basis for the qualitative analysis while all nineteen projects were included in the COCOMO II based analysis of the key factors influencing productivity. The study also set out to identify what mediating factors alter the influence of ethnic diversity on the productivity of software teams. The increasing number of ethnically diverse teams brings opportunities and challenges which require software development project managers to understand and deal with diversity effectively.

Overview of the Thesis

Chapter one provided an introduction to the study, presenting the background and the research questions. This was followed by a summary of the research design and an explanation of why this study is significant. Chapter two presented a review of the literature relevant to this study, beginning with a discussion of the importance of computer software in today's society and the demands for more efficient software production. The factors that have been found to affect productivity were examined, including the personnel and team related factors such as team cohesion, composition and diversity. This led to a discussion of the influence of ethnic diversity on team performance and a review of previous research into diversity in software development teams. Organisational context was shown to affect the outcomes arising from diversity and key differences between public and private organisations were highlighted. As ethnicity is a critical aspect of diversity which has been shown to affect team performance and outcomes, the definitions and dimensions of ethnicity were presented, along with the different cultural values associated with ethnic diversity. Some of the effects of ethnic diversity previously researched include the impact on innovation, conflict and communication. The final section of the chapter examined how productivity is measured, and existing models of software development productivity including the Constructive Cost Model (COCOMO II) which is used in this study as the basis for the conceptual model.

Chapter three explained the mixed methods design and the rationale. The theoretical framework was outlined, including the relevant theories which contributed to the development of the conceptual model. The sample selected for this study was described, including a summary of the software development projects investigated. This was followed by a description of the instruments used in this study and the data collection process. An explanation is provided of how validity, legitimisation and transferability have been addressed. The ethical considerations were then presented and how they have been addressed, along with the pretesting that was performed. Finally an explanation was provided of the analysis approach used to bring together the results.

Chapter four presented the qualitative results, reporting on the major factors affecting productivity and the influence of ethnic diversity. Those factors that influenced productivity within the ethnically diverse software development teams were analysed, beginning with the most frequently cited factors. The influence of ethnic diversity on each of the major productivity factors was also discussed where there was evidence that ethnic diversity was important. Those factors that altered the influence of ethnic diversity were explored and the chapter concluded with an examination of the latent or underlying themes implicit in the project managers' responses.

Chapter five began with the descriptive statistics for the data which combines COCOMO II productivity variables with team diversity indices. This provided an overview of the data and confirmed its suitability for the analysis methods employed. The factors correlated with productivity were then analysed as well as the correlations between ethnic diversity and productivity. Examining these associations provided information about how ethnic diversity was associated with the productivity of software development teams. As government projects had significantly different characteristics from non-government projects, the correlations between ethnic diversity and other variables were analysed within these two groups of projects separately.

Finally, this chapter describes the synthesis of the qualitative and quantitative results, to elucidate and present meta-inferences arising from the mixed methods approach. This begins with a comparison of the major findings from each method and is followed by an analysis of the major areas of convergence between the two sets of findings. The key findings relating to the research questions are summarised and the strengths and limitations of the study considered. The chapter concludes the thesis by commenting on the implications for theory and practice and finishes with recommendations for further research.

Synthesis of Results

An examination of the qualitative and quantitative results demonstrates the strength of the mixed methods approach in achieving a degree of completeness in seeking to answer the research questions. The qualitative results provide insights into how team processes are influenced by ethnic diversity in software teams while the quantitative results report on the association between ethnic diversity and productivity. While the qualitative results provide useful information that enables examination of the workings of the ethnically diverse software teams, they do not present an overall direct relationship between ethnic diversity and productivity. These differences in the findings from each method are illustrated in Table 21 which presents a comparison of the findings based on groupings related to the research questions.

Table 21. Comparison of qualitative and quantitative findings

	Qualitative Findings	**Quantitative Findings**
Factors associated with ethnic diversity	Ethnic diversity led to: • Diverse perspectives • Different ways of working • Something interesting to talk about • Language challenges • Communication barriers • Perceived as unbiased	Ethnic diversity was associated with: • Large teams • Complex projects • Large software products • Longer projects • Higher hours of effort • Poor productivity
Mediating team processes	• Communication • Team cohesion • Strong relationships • Co-located project team • Social events and open invitations • Trust within the team • Shared goals • Agile practices • Planning and Design	Use of software tools for communication and collaboration
Software development productivity factors	Productivity was influenced by: • Complexity • Requirements volatility and quality • Team capability • Application Experience • Attitude and motivation • Communication • Team cohesion • Agile practices • Planning and Design	Productivity was associated with: • Complexity • Requirements volatility and quality • Team size • Software size • Project length • Hours of effort
Contextual factors	• Large organisations were more likely to have complex software development projects	• Government or private sector • Whether the organisation undertaking the software development was primarily a software developer

The qualitative and quantitative results reveal different aspects of the influence of ethnic diversity in software development teams. While there is some convergence in the two sets of results, they also show different aspects of ethnic diversity that influenced or were associated with software development productivity. The source of this divergence is the contrasting perspectives on ethnic diversity and productivity factors that arose from the project managers' views and the project metrics based primarily on the documentation. These two perspectives can be summarised as objectivity versus subjectivity and reflect the epistemological duality of mixed methods research (Harrits, 2011; A. S. Lee & Hubona, 2009). On one hand, the detailed rich view provided by the interviews gives insight into the team dynamics in ethnically diverse software development teams. On the other hand, the objective broader views provided by quantitative measurement yields information about how ethnic diversity is associated with productivity and other factors. Both perspectives reveal information about ethnic diversity in software development teams and are appropriate for this area of study which combines sociological considerations (for example, ethnic diversity and team work) with quantified outcomes (for example, productivity). In this way the differences in the findings are complementary in nature and this allows a holistic set of answers to the research questions. Despite these contrasting perspectives, there are still three major areas of convergence between the results. These are complexity, communication as a mediating factor and requirements quality.

Complexity

There was convergence between the qualitative and quantitative results in showing that complexity in software development projects was associated with poor productivity in the ethnically diverse teams studied. The correlation between productivity and complexity in the quantitative results demonstrated that more complex software projects were less productive. More complex tasks take more effort to complete and this relationship has been observed in software development projects in previous research (Boehm, et al., 2000a; Heijstek & Chaudron, 2009; K. Maxwell &

Forselius, 2000). In the interviews, project managers described complex business requirements and cited complex subtleties to the requirements as the factors that negatively impacted productivity. They also discussed complexity as a result of interdependencies with other projects. This arose particularly in large organisations where there were hundreds of software projects underway at the same time and there were many systems with numerous interfaces between them. As these projects and systems became intertwined this created complexity which negatively impacted productivity.

The software being developed has become increasingly complex over recent decades (Santos & Moura, 2009) making it more difficult for software development productivity to improve. Software projects are growing larger in size and becoming continually more complicated in order to fulfil society's needs (Boehm, 2006) raising the question of how the complexity of software projects can be reduced. One approach raised in the project manager interviews was to break large projects down into smaller projects in an attempt to reduce the amount of complexity each project team needs to deal with. However, this was not always possible and it can also create further complexities as it gives rise to the need for different developers to integrate their work into a single software solution (Boehm, et al., 2000a). For example, where two developers are making different changes to the same program, they subsequently need to merge their changes which can be complex. Such activities rely on communication and co-ordination across the team and between teams which can be face-to-face or via a software tool to support these activities. Such collaboration tools were used on some projects to manage such communication and this is another area of convergence in the results discussed later in this chapter in the section *Communication as a Mediating Factor*.

Complexity in software development projects can be either reduced or dealt with through innovation and problem solving (Cusumano, Crandall, MacCormack, & Kemerer, 2009; Fraser, 2009; G. Lee & Xia, 2010). One of the benefits of ethnic diversity in teams is the possibilities it creates for problem solving. Innovation is closely related to problem solving but it describes the process of creating something new and this distinguishes it from problem solving. Innovation in software development can occur in the three major aspects of software development - product, people or process (which includes how the project is run) (Aaen, 2008; Tidd & Bessant, 2011). In the projects analysed, there was evidence of innovation in the software product being developed and the process used, and this linked in various ways to the ethnic diversity within the teams. Innovation and problems solving were important in the software projects analysed given that a high level of complexity often caused challenges and this was a major factor affecting productivity.

There was evidence in the interviews that problem solving in the software projects was improved by the presence of ethnic diversity in the teams. The type of problem solving that was enhanced by diversity was connective thinking which is where ideas from diverse or unrelated areas are brought to bear on a problem (Post, et al., 2009). This occurred where ethnically diverse team members were co-located and could share ideas as they arose. Connective problem solving also occurred in situations where the project teams were open to sharing their ideas and experienced a high degree of trust.

Sequential problem solving (Erdogmus, 2009; Page, 2007; Post, et al., 2009) was also evident in the interviews with many project managers describing the project team encountering an obstacle and working together methodically to overcome it. This approach to problem solving involved group discussions, detailed feedback from project team members on the challenges encountered and simply persisting in asking more questions of other team members. Problem solving using sequential approaches was less obviously connected to ethnic diversity than connective problem solving. Instead, sequential problem solving was characterised by individualistic approaches rather than collectivist. This collectivist cultural value is associated with some of the ethnicities on the teams (for example, Indian, Chinese and Filipino) more than others (for example, American or European) (Hofstede, et al., 2010). It appeared from the interviews that teams with greater ethnic diversity, combined with an openness to share ideas and approaches were more likely to adopt a collectivist and more successful approach to problem solving rather than individualistic.

In some of the ethnically diverse teams there was evidence of innovation in terms of the software product being developed. However, in order for this innovation to occur, it was clear there had to be an environment where all team members felt they could ask questions and raise ideas without risk of being made to feel inadequate. The importance of trust, relationships and communication were highlighted as important factors enabling and supporting the innovation process. Team members in some projects were encouraged to question and challenge design ideas being proposed when they felt the solution design was not correct and this led to more successful solutions and improved overall productivity. This challenging of ideas could only occur where team members felt comfortable to communicate ideas honestly and openly, as this helped remove barriers to communication. Previous work has found that a team with diverse values, arising from cultural diversity, can make members of a software team less likely to co-operate and speak up unless teams are effectively managed (Liang, et al., 2007). These previous findings, when combined with the results from this study show there is a tension that exists between ethnic diversity causing barriers between team members and the benefits of innovation and problem solving that can arise from diverse perspectives to help address complex challenges. Given that higher complexity was associated with poor productivity in the projects studied, this suggests that ethnic diversity did not fully mitigate the negative effect of complexity on productivity. However, the degree to which ethnic diversity could help overcome the negative effect of complexity on productivity was shown in the interviews to be largely dependent on the environment created and how the project manager addressed team cohesion.

Communication as a Mediating Factor

There was corroboration between the quantitative and qualitative results regarding the importance of communication as a mediating factor as it altered how ethnic diversity influenced the productivity of the software development teams. In the interviews it was revealed that communication was affected by ethnic diversity, but also enabled the benefits of ethnic diversity to occur. Effective communication within the team helped develop relationships, build trust and improve team cohesion. This allowed different perspectives and ideas to be raised and discussed in a supportive team. The presentation and evaluation of diverse ideas associated with ethnic diversity helped solve difficult problems on the software projects and led to innovations which allowed the software required to be developed more productively.

The quantitative results showed that ethnically diverse software teams made greater use of software tools to support communication within the team and project managers reported that this helped enhance communication in ethnically diverse software teams. The use of communication and collaboration tools in software projects has previously been shown to improve productivity in software projects (Baik, Boehm, & Steece, 2002; Candrlic, et al., 2006; Tiwana, 2008). The use of tools to support communication and collaboration within the software teams was also raised in the qualitative results when discussing the use of agile practices. An agile collaboration tool was specifically identified as an enabler for success by project managers. For example on one agile project a physical "scrum board" was used to show the progress of each component being developed. This was complemented by an electronic repository which held the detailed information about each change and its status. The combination of these two tools helped team cohesion and communication. The use of computer mediated communication tools has previously been found to be one way of addressing communication challenges that can exist within ethnically diverse software teams (Nam, et al., 2009; Shachaf, 2008). Therefore, there was corroboration in the results showing that the use of software development tools was a mediating factor that helped facilitate the benefits of ethnic diversity in software development teams.

Requirements Quality and Volatility

Requirements quality and volatility was the third most frequently cited factor in the interviews affecting productivity. The requirements are the documents that specify what the software must do and there were two key ways in which requirements affected productivity that were discussed by project managers. One was low quality requirements resulting in the system developed being not fit for use. This created the need to make changes to the system, increasing the overall effort required to deliver the software. The other was the amount of change in the requirements during the project. Where requirements changed during the project this negatively impacted productivity as more effort was required to accommodate changed requirements.

The quantitative results also showed that projects where the requirements changed significantly and frequently, had worse productivity as the higher the level of requirements volatility, the more hours of effort were required to produce each unit of software. Requirements volatility was not correlated with the number of ethnicities in the team, but as most of the teams analysed had a high degree of ethnic diversity, this is a factor which may influence productivity in ethnically diverse software teams. That is, in the software projects studied, if the requirements changed significantly and often during the project, then this negatively impacted productivity.

When requirements changes were carefully managed the negative impact on productivity was mitigated. Requests from the customer to make major changes to requirements were discussed but deferred to a later project. Although minor requirements were added, the major changes were successfully deferred after some negotiation. Managing requirements is an important component of the project management discipline (Hartley, 2008; Pinto, 2009; Project Management Institute, 2008). However requirements volatility also occurred where requirements were not correctly gathered in the first place. Where there are issues with the requirements capture process then the productivity of a software project will be impaired (Cerpa & Verner, 2009; Chua & Verner, 2010; Hofmann & Lehner, 2001).

Requirements gathering is primarily a process of interaction and relies on interpersonal relationships. Therefore relationships, trust and communication are important between the customer or end user, and those gathering the requirements. As shown in the qualitative results, these team processes are all influenced by ethnic diversity, and depending on the team environment, can either positively or negatively impact interpersonal activities. As the teams studied were predominantly ethnically diverse, and requirements volatility was significantly correlated with poor productivity, when this is combined with the interview responses, it is highly probable that this factor is important in the productivity of ethnically diverse software development teams.

Summary of Synthesis

Synthesis of the major findings from the qualitative and quantitative results shows some degree of convergence, as well as providing an understanding of the influence of ethnic diversity in software development productivity. Convergence was highest with the software development productivity factors, where complexity and requirements quality were key factors in both the quantitative and qualitative results. Some degree of convergence was evident in the mediating team process of communication, with different types of communication appearing in the two sets of results. The divergence of the results was examined in order to consider the completeness of the findings. The qualitative results provided insight into direct consequences of ethnic diversity in software teams and the team processes that altered how ethnic diversity influenced productivity. In contrast, the quantitative results report on the association between ethnic diversity and productivity which represents an indirect relationship mediated through aspects not revealed in the quantitative results. While this meant the results from the two methods to some degree diverged, they are also complementary in nature.

The complementary nature of the results was synthesised to focus on the connections between the most significant themes identified in the qualitative and quantitative results. The quantitative results showed that large and complex projects were less productive and these projects were most affected by the major productivity factors, communication and team cohesion, identified in the qualitative results. The quantitative results also showed that large or complex projects had greater ethnic diversity and the synthesis indicated large projects were influenced by ethnic diversity more so than small projects. However, the qualitative results showed that the potential benefits of ethnic diversity were greatest in large or complex projects and were more likely to occur where the project environment engendered trust and supported the building of strong relationships.

The quantitative finding that government projects were significantly less productive than non-government projects led to an evaluation of the productivity drivers between the two sectors. This has been used to complement the qualitative comparison of government and non-government projects as it provides further information about the source of the differences in productivity. Communication was the only factor that was amongst the top three productivity drivers identified by both government and non-government project managers. However, the focus differed between the two groups of project managers with the government ones concentrating on communication within the team and non-government project managers talking more about external communication. Ethnic diversity-related themes were identified far more often in the government projects with non-government project managers generally viewing ethnic diversity more positively.

Having presented the synthesis of the qualitative and quantitative findings, the following section applies the full set of findings from the qualitative analysis, quantitative analysis and synthesis to address the three research questions.

Answers to Research Questions

Question 1. What factors influence productivity in ethnically diverse software teams?

The factor that project managers believed influenced software development productivity the most in ethnically diverse projects was communication. The aspects of communication that were identified as being critical to the productivity of the software development projects were timeliness, the mode of communication and the use of feedback. The key measure of good or effective communication was whether the correct message was received. Some of the communication issues were within project while others related to boundary spanning exchanges with stakeholders, other teams, customers, vendors and project sponsors. Face-to-face communication was generally seen as enhancing the effectiveness of communication, as was the use of agile software development practices. Effective communication within teams helped develop relationships, build trust and improve team cohesion.

Team cohesion was also widely cited by the project managers as influencing productivity in the ethnically diverse software development teams. Good team cohesion enhanced relationships and improved trust within the team, enabling open communication and supporting freer discussions amongst team members. This provided the environment in which diverse ideas and approaches to solving complex software development problems could be surfaced and discussed. This was important to improving productivity as complexity in the software development projects was another major factor influencing productivity according to both the qualitative and quantitative results. Complexity arose due to multifarious business requirements and subtleties in the requirements which in turn led to poor productivity. Complexity as a result of interdependencies with other projects also had a similar effect. This arose particularly in large organisations where there were hundreds of software projects underway at the same time and there were many systems with numerous interfaces between them. As these projects and systems became intertwined, complexity arose which negatively impacted the productivity of the projects.

Project size was a source of complexity and projects which were larger in terms of team size, project length, total hours of effort and product size were less productive. Complex and large projects also tended to have greater ethnic diversity within the team. The correlations between poor productivity, more ethnicities and large projects was due to a larger team with more people being more likely to include a broader range of people, including different ethnicities. Communication challenges accounted for, at least in part, the association between ethnic diversity, large projects and poor productivity.

Poor quality requirements and highly volatile requirements both influenced productivity. Poor quality requirements resulted in the system developed not being fit for use. This led to the need to make changes to the system, increasing the overall effort required to deliver the software. Where requirements changed during the project, productivity was negatively impacted as more effort was required to accommodate changed requirements. Requirements volatility such as this also arose where requirements were not correctly gathered in the first place.

Requirements gathering is primarily a process of interaction and relies on interpersonal relationships. Those involved in this activity were of a variety of different ethnicities. This study has shown that in ethnically diverse software teams, requirements quality and volatility was a factor that influenced productivity. Furthermore, the degree to which relationships, trust and team cohesion were cultivated within the ethnically diverse teams had an impact on the effectiveness of requirements gathering, and this in turn influenced software development productivity.

Question 2. Does ethnic diversity influence the productivity of software teams?

There was evidence that problem solving in the software development projects was improved by the presence of ethnic diversity in the teams where dissimilar views or ideas were brought to bear on a problem. Project managers reported that productivity was improved through having diverse perspectives which they believed were enhanced by team members' ethnically diverse backgrounds. This meant that improved decision making was useful in balancing out any biases in the team. Innovation was also enhanced, which helped address complexity and contributed to improved productivity. Connective problem solving also occurred in situations where the project teams were open to sharing their ideas and experienced a high degree of trust. Teams with greater ethnic diversity, combined with an openness and willingness to share ideas and approaches tended to adopt a collectivist and more successful approach to problem solving. This occurred where ethnically diverse team members were co-located and could share ideas as they arose.

The results of this study show that while greater complexity had a negative influence on the productivity of ethnically diverse software development teams, the ability to innovate and solve problems could be enhanced, in certain circumstances, by the ethnic diversity within the team. Ethnic diversity helped improve innovation, but generally only when measures were taken that enabled all team members to contribute.

The variety of backgrounds and experiences within the teams meant that team members were interested in finding out about each other and this acted as a catalyst for initiating and maintaining relationships between them, particularly in informal settings. Team members viewed the fact that others were different from themselves as a positive aspect. This positive effect on relationships contrasts with the findings of previous research (Ayub & Jehn, 2011; Jehn, et al., 1999; Sujin, 2005) which found that team diversity had a negative impact on relationships with individuals seeing dissimilar others as outside of their group, leading to communications challenges and conflict due to disparate values.

The potential for ethnic diversity to improve relationships was most significant where an environment of trust was created by the project manager. Opportunities for relationship building needed to occur or be created where interest in other team members' background and experiences could be explored and used to strengthen team connections. In this way the project manager plays a key role in facilitating relationships in any project team, but especially in ethnically diverse teams where there may be higher risks of social categorisation. It was apparent in most of the projects studied that project managers played a crucial role in improving team cohesion in ethnically diverse software teams by providing an environment where relationships could form and develop.

Language challenges, generally in terms of poor English, were a diversity-related factor negatively impacting communication, which in turn affected productivity. This affected written communication, face-to-face communication and talking on the telephone. New Zealand is an English speaking country and the main impact of ethnic diversity identified by project managers related to problems with English. . Language challenges were compounded on large projects which led to an increase in the time spent communicating across large teams. Where teams were geographically dispersed, people would dial in from different locations and in this situation language challenges were an issue.

Ethnic diversity led to communication barriers on some projects where hierarchies existed between ethnic subgroups, for example, where those of a lower caste would not speak up and instead deferred to those of a higher caste. These hierarchical issues represent a form of social categorisation and the results are consistent with previous work which finds negative impacts of ethnic diversity arising from social categorisation (Dahlin, et al., 2005; van Knippenberg, et al., 2011). This type of issue requires understanding by the project manager for it to be successfully managed and to prevent it negatively affecting team productivity.

Barriers between people seen as in-group or out-of-group also had an influence on communication and therefore team productivity. The communication problems reported in some of the ethnically diverse teams were underpinned by intergroup differences as team members were more likely to acknowledge and consider communication from in-group members (in this case, of the same ethnicity) rather than out-of-group members. These communication challenges had a negative influence on software development productivity as communication was critical for productive software development teams.

Question 3. What mediating factors alter the influence of ethnic diversity on the productivity of software teams?

Three areas of mediating factors altered the influence of ethnic diversity on productivity. The first was communication techniques and tools implemented by project managers that helped the benefits of ethnic diversity to occur. These included varied mediums and forums, face-to-face contact and the use of computer based communication tools. The use of agile practices, particularly those relating to communication within the team also helped the benefits of ethnic diversity to occur. The second area of mediating factors was the level of trust within the team, the strength of relationships and the degree of team cohesion. Project managers enabled these mediating factors to occur by co-locating the project team and through social events or other opportunities to connect and remove potential barriers between groups. The third area encompasses factors which altered the influence of ethnic diversity but were outside the control of the project manager, such as the size of the project and whether the organisation was part of the government or private sector.

Actions which project managers took created an environment that supported effective communication and enabled the benefits of ethnic diversity such as diverse perspectives to help the team be productive. Where project managers took measures to facilitate team communication and develop team relationships, this had a positive influence on productivity in ethnically diverse teams. A variety of approaches to communication were adopted such as having informal stand up meetings each morning, more formal weekly meeting and occasional social gatherings. These varied forums for communication within the team supported effective information exchanges and enabled team members to communicate in the form with which they were most comfortable. By getting people together in an effort to break down communication barriers, project managers were able to overcome negative influences and enable the benefits of ethnic diversity to occur. The use of computer based communication tools were also identified in the results as helping mitigate the effects of language related challenges in ethnically diverse software development teams.

Project managers described how they developed trust within their teams and how this helped break down barriers that existed between team members. Trust is a critical enabler for effective communications within the project team as well as being important for achieving team cohesion. As an ethnically diverse team requires individuals to interact with others who may hold different values, beliefs and ways of doing things, it was essential that the project manager establish an environment of trust between team members. Project managers described steps such as seating team members together, arranging informal social gatherings and ensuring all views could be heard in order to help develop trust within the team. These mediating factors were seen to improve team cohesion, improve productivity and enable the benefits of ethnic diversity to occur.

As well as mitigating any negative effects of ethnic diversity in their teams, developing an environment of trust also contributed to building strong relationships within the team. When the project manager created a suitable environment, the diverse backgrounds of an ethnically diverse team helped generate interest and informal conversations between team members. This aided communication and team cohesion, the two productivity factors most cited by project managers. Furthermore, opportunities for relationship building needed to occur or be created where interest in other team members' background and experiences can be explored and used to strengthen team connections. In this way, the actions taken by the project managers to create an environment of trust and facilitate relationships within ethnically diverse teams were an important mediating factor that determined whether ethnic diversity had a positive or a negative effect on productivity.

Assigning the right people to the right roles is another way project managers were able to make the most of ethnic diversity within their teams. The effectiveness of project managers in allocating team members to the right role was a mediating factor that determined whether diversity-related aspects such as language challenges or communication barriers negatively affected productivity. For example, a software developer whose English was not fluent may have relatively little need to communicate widely and therefore this factor has no negative impact on productivity.

The project size was another mediating factor that altered the influence of ethnic diversity on software development productivity. Larger project teams are more likely to have more ethnic groups and are more likely to have poor productivity. However, small projects with ethnic diversity (that is, more ethnic groups as a proportion of the overall team size as opposed to simply more ethnic groups) are more likely to have good productivity. The factors that cause large projects to be less productive (such as poor communication) can be exacerbated by ethnic diversity. The increased volume and complexity of interactions on large projects amplifies the effect of communication barriers, language challenges and incompatible ways of working that can all inhibit working together. Therefore, the size of the project affects whether ethnic diversity influences overall software development team productivity.

Finally, whether the project was undertaken in the public or private sector was an important contextual factor that altered how ethnic diversity influenced productivity. Seven of the projects studied were government projects and twelve were private sector. Within the private sector, ethnically diverse project teams were more likely to be productive than those in the public sector. There were notable contrasts in the way ethnic diversity was perceived by project managers in the two sectors. Government project managers identified ethnic diversity as having a negative effect on team cohesion as people from a different country had a different way of working which hindered the team working together effectively. In contrast, on non-government projects the project managers described how team members had different but complementary work styles. The non-government project managers perceived that the different types of roles on the software development projects required diverse working styles. This meant that it was seen as an advantage to have people with different ways of working associated with ethnic diversity. Although communication barriers were discussed by government and non-government project managers, the issues were only associated with ethnic diversity in the government projects.

Strengths of the Study

The mixed methods approach used in this study provides answers to the research questions from different perspectives and these have been combined to provide a broader understanding of the results. The qualitative results give greater insight into the team dynamics and direct consequences arising from ethnic diversity in software teams. Furthermore, the qualitative approach has enabled an examination of the team processes that mediated how ethnic diversity influenced productivity. In contrast, the quantitative results report on the direct and indirect associations between ethnic diversity and productivity. Both perspectives reveal information about ethnic diversity in software development teams and are appropriate for this area of study which combines sociological aspects, such as the team dynamics arising from ethnic diversity, with productivity which is a quantifiable outcome of software development. In this way the differences in the findings are complementary in nature.

This study has brought together formerly disparate strands of work as ethnic diversity and software development productivity have not previously been analysed together. Combining these two concepts has important implications for practice as new and innovative approaches to improving productivity are significant objectives for many software producing organisations.

The primarily qualitative nature of the study and the interviewing of project managers as key informants who are familiar with their team allowed the research topic to be analysed in depth. The perceptions and insights of the project managers are important as they tapped into their previous experience of software projects when describing the influence of ethnic diversity on the software teams. The nuanced nature of how ethnic diversity affects team dynamics, and therefore productivity, cannot be measured using quantitative data alone. The interviews enabled the surfacing of subtleties in the reasons why ethnic diversity influenced productivity and provided unexpected avenues of explanation. This has led to greater richness and depth in the results and enhanced the explanation quality of the study.

Using the qualitative and quantitative results to corroborate and confirm the findings from each method increases the credibility and validity of the results. Synthesis of the results has enabled the examination of whether the results from one method converge with the results from another method and consideration of where there was divergence. This involved combining the results from two different sources of data (interviews and project documentation) and analysing the areas of convergence and divergence. Mixing of qualitative and quantitative methods in this way has enhanced the degree of confidence in the findings and provides more certainty about the implications.

Limitations of the Study

This research has limitations relating to transferability, sample size and bias. First, the study was undertaken in one city only, Wellington, New Zealand. Therefore the sample is not necessarily representative of all of New Zealand or the global software industry and the findings may therefore reflect characteristics that are specific to Wellington. This could restrict the ability to draw conclusions about the transferability of the findings to other similar contexts. However this limitation has been mitigated by the in-depth nature of the investigation that involved multiple data sources and a description of the characteristics and context of each project (see *Appendix I – Data for All Variables for All Projects*) The capture and reporting of the context of each project helps to analyse the scope of transferability (Walsham, 1995) but is balanced with the requirement for participating organisations to be unidentifiable.

Second, only a limited number of software producers were willing to participate. Studies of software development productivity are typically conducted using quantitative methods but gathering a sample of software development projects which is sufficiently large and representative in order to achieve generalisable results is challenging. These challenges arise because of obstacles to obtaining access to project data and differences between the types of data captured by different organisations (K. Maxwell, 2001; Mendes & Lokan, 2008). Such challenges have been encountered in this study but this situation was largely anticipated during the planning of this study and contributed to the decision to adopt a mixed method design involving both quantitative and the dominant qualitative methods. Combining methods to "compensate for the weaknesses in one approach by using the other" (Venkatesh, et al., 2013, p. 6) is an acknowledged and valid reason for using mixed methods.

The sample size was relatively small, when considering the high number of productivity factors captured. Further, the study involved a convenience sample, given the challenges in recruiting software producers to participate. This is a widely acknowledged challenge to undertaking detailed research of software development projects (Mens, et al., 2005; Shepperd & Cartwright, 2001; Verner, et al., 2009). Using a larger sample which was randomly selected would increase the likelihood that the findings were representative of the software development industry, particularly for the quantitative component. However, this research was not solely a quantitative study and the qualitative findings revealed in-depth insights into project managers' perspectives and the team dynamics of ethnically diverse software teams which would be less likely to surface using a purely quantitative approach.

The researcher's experiences and beliefs will have inevitably influenced several key decisions regarding the scope and design of this study. For example, the choice of research domain, the selection of software producers invited to participate and the nature of probing undertaken by the researcher during interviews are all undoubtedly affected by the researcher's own experiences. The researcher in this study is an experienced software professional who has worked in a variety of roles over the previous 20 years. These have included software project manager, test manager, business analyst and software developer. While this may have presented a particular bias to the study, this is balanced by the advantage of the researcher having an intimate knowledge of the processes of software development, thereby “extending the range and sensitivity of human sensing” (Betz, 2010, p. 66).

Measures were taken in both the design and execution of the study to balance the inherent researcher subjectivity with the need for credible and objective findings. The research design was reviewed in response to feedback from academic peers through conference presentations and publications. This included doctoral symposiums, both in New Zealand and Australia. The interviews were designed to enable the interviewee to tell their own story with the use of open questions. Interview transcripts were provided to interviewees for their review and the analysis of the data was validated through pre-testing and refinement involving academic staff.

Project managers, as the key informants used for this study, could be viewed as not representing the views of the whole team or other stakeholders and participants. However project managers have extensive interaction with their team, they play a key role in determining the project approach (Sebt, et al., 2010) and have a good understanding of the factors that affected the productivity of a project (Ehsan, et al., 2010; Wang, 2009). Team composition is often determined, or at least influenced by project managers (Sebt, et al., 2010). For these reasons, the interviews with project managers, combined with quantitative data gathered from project documentation provided the breadth of data required to answer the research questions.

Contribution

Implications for Theory

Bringing together the mediating factors discussed builds on earlier work by the Diversity Research Network (Kochan, et al., 2003) to develop a theoretical model of the influence of team diversity on performance. This model includes organisational context and team processes which are both factors that alter the influence of team diversity on productivity. While the original model from Kochan, et al, includes different types of team diversity and is not specific to one industry, the results of this study have resulted in a specialised version of the model (see *Figure 7* adapted from Kochan et al (2003)) showing ethnic diversity in software development teams influencing productivity via a series of mediating factors.

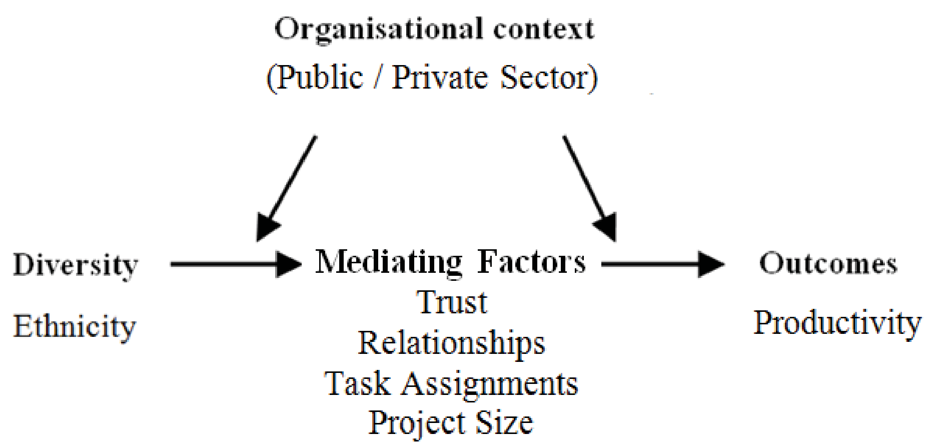


Figure 7. Model of the influence of ethnic diversity on team outcomes, adapted from Kochan, et al. (2003)

This study contributes to the mixed methods literature by producing meta-inferences and thereby going some way to reduce the "contribution shrinkage" (Venkatesh, et al., 2013, p. 11) that occurs when papers from mixed methods studies are published based on individual methods with no synthesis. Examination of diversity in software teams has previously been studied either using only quantitative (for example, Egan, et al., 2006; Liang, et al., 2007; Pieterse, et al., 2006) or qualitative methods (for example, Borchers, 2003; Shachaf, 2008; Walsham, 2002). This answers the call for greater use of mixed methods in IS research (Mingers, 2001; Venkatesh, et al., 2013) as the approach provides a more holistic understanding of complex systems such as the software development process.

A further contribution of this study is to the body of literature that examines software development as a socio-technical system (STS). STS theory has previously been used to analyse a number of different aspects of software development but this research of the sociological effects of ethnic diversity in software projects represents a unique examination. Its uniqueness is in part because it is the first mixed methods research into the influence of ethnic diversity in the productivity of software development teams, and also because it takes into account other factors already known to affect software development productivity.

Implications for Practice

The findings indicate that relationship building is important for improving productivity in ethnically diverse teams and this has implications for the practice of managing software projects. Where project managers can create situations, events and an environment that provides opportunity for strengthening team connections, then ethnic diversity can help improve team productivity by helping to build effective work relationships within the software team. This is particularly important in software development teams where a high degree of interaction is required to formulate solutions, solve complex problems and co-ordinate activities.

This study has shown how ethnic diversity in software development teams influences communication and relationships, and thereby affects team cohesion. Team cohesion is an important productivity factor in software development and team performance in general but one criticism of the software estimation model COCOMO II is that team cohesion is not accurately understood (Snowdeal-Carden, 2013). While the Snowdeal-Carden study assessed quantitative instruments for measuring team cohesion in software development teams, this study uses a qualitative approach to understanding the underlying influencers of team cohesion. These findings have implications for assessing team cohesion and estimating its effect on software development productivity using COCOMO II or other estimation methods.

Where there is ethnic diversity within software development teams and communication challenges may arise, frequent and varied forums for communication enable effective information exchanges. If project managers use a variety of methods to facilitate team communication, then the team is likely to be more productive. Many agile based communication practices provide the tools required to support effective communication. For the frequent transactional communication, agile practices such as daily stand-up meetings, ad-hoc informal discussions and generally using face-to-face communication are approaches which a number of project managers cited as improving communication outcomes. The use of computer based collaboration tools provides another communication channel which can help mitigate potential communication barriers in ethnically diverse software development teams.

Finally, this study shows that ethnic diversity can provide benefits to software development projects provided it is managed appropriately. Ethnic diversity enables different perspectives to be brought to bear on software development challenges in order to arrive at the best solution and improve overall team capability. A variety of working styles can also be associated with ethnically diverse teams and this can be best utilised by ensuring each person is allocated to the right role. Software development projects typically require varied but complementary roles to be fulfilled and in this way diversity within a software team is valuable. These potential benefits of ethnic diversity should be considered when determining the composition of software development teams. Furthermore, where software development teams are ethnically diverse, it is important for project managers to give additional consideration to ways of enabling team cohesion and facilitating communication to ensure all views are considered. Given that many managers currently have little or no understanding of how to deal with diversity (Korn / Ferry Institute, 2013) specialist training and support would enable software development project managers to improve the performance of their teams.

Future Research

Based on the results of this study regarding the influence of ethnic diversity in software teams, five areas for future research are presented. The first area is that the findings from this study on ethnic diversity could be confirmed and explored further through research which uses interviews from a wider range of project team members and includes observation of projects in progress. Observation of critical points of interaction between team members such as project meetings and ad-hoc conversations may reveal greater understanding of the influence of ethnic diversity on communication and team cohesion in software teams. A case study approach could be used with a strong ethnographic component in order to more fully explain the social interactions that occur in ethnically diverse software development teams.

The findings were based solely on software projects undertaken in Wellington, New Zealand. Future research could study software projects from other cities and examine the influence of ethnic diversity in software projects in other locations. Such replication could serve to strengthen the findings from this study or highlight differences by location. In this way, the scope of transferability or generalisation could be further elucidated by replication logic whereby the findings from one study are replicated in other situations (Eisenhardt & Graebner, 2007; Yin, 2009).

Future research should examine the influence of ethnic diversity in a wider range of software project success factors. While productivity is an important and measureable aspect of software projects, a number of other important aspects include quality, customer satisfaction, team satisfaction, on time delivery and the accuracy of estimates. Alternative measures of software development productivity could also be used to complement the function point based measure of project delivery rate (PDR) used in this study. Although the PDR is widely used and is a core measure in an industry benchmark database of over 5000 software projects (International Software Benchmarking Standards Group, 2009), this functionality based measure cannot be used on software projects where there is no change to end user functionality. Considering the influence of ethnic diversity using other measures of productivity or project success would be valuable in understanding whether the findings of this study can be replicated using alternative perspectives on software projects.

Based on the findings of this study, there is evidence to support further investigation into the influence of team composition, particularly ethnic diversity. The results suggest that the influence of ethnically-diverse teams is amplified on large, long and complex projects and that a diversity index could potentially be included in COCOMO II as a scale factor as diversity appears to have a non-linear relationship with productivity. Furthermore, as far as can be ascertained, there has not been any prior research capturing data for the COCOMO II variables for software projects in New Zealand. Therefore, the data from this study provides the ability to calibrate the COCOMO II model to the New Zealand software development industry.

Finally, this study revealed that the actions of project managers can be a significant determinant of whether the benefits of ethnically diverse teams are realised. Therefore, future research into how project managers influence the outcomes of software projects would be valuable in understanding what actions project managers should take. While some useful actions have been identified in this study such as building trust and relationships within the team, co-locating team members and allocating people to the right roles, there may be other areas of focus which are important in ensuring the benefits of ethnic diversity can occur. This could include interviews or surveys with project team members to assess their perceptions of whether the actions taken by project managers enabled or inhibited the benefits of ethnic diversity such as innovation and problem solving to occur.

Summary of the Conclusions

While ethnic diversity has been examined in many types of teams, as far as can be ascertained, this is the first mixed methods study to report on the influence of ethnic diversity in software development productivity. Therefore, this study is significant as it examines how ethnic diversity influences the productivity of software teams and reports on what factors help make ethnically-diverse software teams more (or less) productive. The findings show that ethnic diversity can improve software development productivity if project managers are able to create an environment of trust and enable relationships to develop within the team. With such an approach, communication challenges can be overcome along with the use of appropriate communication tools.

The study differs from previous studies of software development productivity in that it involves analysing projects completed by New Zealand organisations using a mixed methods approach. As far as can be ascertained, no studies have examined these aspects in a New Zealand context in this way. For this reason, it is expected that the results will be of particular interest to organisations in New Zealand, when considering the composition of software development teams and managing those teams.

REFERENCES

Aaen, I. (2008). Essence: facilitating software innovation. *European Journal of Information Systems, 17*(5), 543-553.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). *New directions on agile methods: A comparative analysis*, Portland.

Abran, A., Desharnais, J.-M., Oligny, S., St-Pierre, D., & Symons, C. (2009). *The COSMIC Functional Size Measurement Method Version 3.0.1 – Measurement Manual*: Common Software Measurement International Consortium.

Addy, E., Mili, A., & Yacoub, S. (1999). A case study in software reuse. *Software Quality Journal, 8*(3), 169-196.

Afsharian, S., Giacomobono, M., & Inverardi, P. (2008). *A framework for software project estimation based on COSMIC, DSM and rework characterization*. Paper presented at the Proceedings of the 1st international workshop on Business impact of process improvements.

Ahmed, M. A., Saliu, M. O., & AlGhamdi, J. (2005). Adaptive fuzzy logic-based framework for software development effort prediction. *Information and Software Technology, 47*(1), 31-48.

Airhihenbuwa, C. O. (2007). *Healing Our Differences: The Crisis of Global Health And the Politics of Identity*: Rowman & Littlefield Publishers.

Albrecht, A. J. (1979). *Measuring application development productivity.* Paper presented at the Joint SHARE/GUIDE/IBM Application Development Symposium 1979, Monterey.

Alesina, A., & La Ferrara, E. (2005). Ethnic diversity and economic performance. *Journal of Economic Literature, 43*(3), 762-800.

Ali, M., & Brooks, L. (2008). *Culture and IS: National cultural dimensions within IS discipline*: Bournemouth University.

Alleman, G., Anderson, D. J., Avery, C. M., Bauer, M., Brosseaus, J., Butler, C. W., et al. (2004). *The Evolution of Agile Project Management.* Arlington: Cutter Consortium.

Allen, I. E., & Seaman, C. A. (2007). Likert scales and data analyses. *Quality Progress, 40*(7), 64-65.

Almaney, A. (1974). Communication and the systems theory of organization. *Journal of Business Communication*, *12*(1), 35-43.

Alvesson, M. (2003). Beyond Neopositivists, Romantics, and Localists: A Reflexive Approach to Interviews in Organizational Research. *The Academy of Management Review, 28*(1), 13-33.

Alyahya, M. A., Ahmad, R., & Lee, S. P. (2009). Effect of CMMI-based software process maturity on software schedule estimation. *Malaysian Journal of Computer Science, 22*(2), 121-137.

American Psychological Association. (2002). Ethical principles of psychologists and code of conduct. *American Psychologist, 57*, 1060-1073.

Anantatmula, V. S. (2010). Project Manager Leadership Role in Improving Project Performance. *Engineering Management Journal*, *22*(1), 13-22.

Anderson, N. R., & West, M. A. (1998). Meauring climate for work group innovation: Development and validation of the team climate inventory. *Journal of Organizational Behavior, 19*(3), 235.

Ang, S., & Slaughter, S. A. (2001). Work outcomes and job design for contract versus permanent information systems professionals on software development teams. *MIS Quarterly, 25*(3), 321-350.

Anthes, G. (2005). The future of IT. *Computerworld, 39*(10), 27-27.

Arrow, H., McGrath, J., & Berdahl, J. (2000). *Small groups as complex systems*. Thousand Oaks: Sage Publications.

Asmild, M., Paradi, J. C., & Kulkarni, A. (2006). Using Data Envelopment Analysis in software development productivity measurement. *Software Process: Improvement and Practice, 11*(6), 561-572.

Atkinson, R., Crawford, L., & Ward, S. (2006). Fundamental uncertainties in projects and the scope of project management. *International Journal of Project Management, 24*(8), 687-698.

Auer, P. (2002). *Code-Switching in Conversation: Language, Interaction and Identity*: Taylor and Francis.

Ayub, N., & Jehn, K. A. (2011). The moderating role of social attitudes in the relationship between diversity and conflict. In M. A. Rahim (Ed.), *Diversity, Conflict, and Leadership: Current Topics in Management* (pp. 13-40): Transaction Publishers.

Badkar, J., & Tuya, C. (2010). *The Asian workforce: a critical part of New Zealand's current and future labour market*. Wellington: Department of Labour.

Baik, J., Boehm, B., & Steece, B. M. (2002). Disaggregating and calibrating the CASE tool variable in COCOMO II. *Ieee Transactions on Software Engineering, 28*(11), 1009-1022.

Banker, R. D., Chang, H., & Kemerer, C. F. (1994). Evidence on economies of scale in software development. *Information and Software Technology, 36*(5), 275-282.

Banwet, D. K., Yadav, S. S., & Momaya, K. (2003). *Management of Research & Development in the New Millennium.* Paper presented at the International Conference on Management of Research & Development (ICMARD 2003), Delhi.

Bar-On, R. E., & Parker, J. D. (2000). *The handbook of emotional intelligence: Theory, development, assessment, and application at home, school, and in the workplace*: Jossey-Bass.

Baresi, L., & Morasca, S. (2007). Three empirical studies on estimating the design effort of Web applications. *ACM Trans. Softw. Eng. Methodol., 16*(4), 15.

Barki, H., & Jon, H. (2001). Interpersonal Conflict and Its Management in Information System Development. *MIS Quarterly, 25*(2), 195-228.

Baron, R., & Kenny, D. (1986). The moderator-mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of personality and social psychology, 51*(6), 1173-1182.

Bass, B. M., Valenzi, E. R., Farrow, D. L., & Solomon, R. J. (1975). Management styles associated with organizational, task, personal, and interpersonal contingencies. *Journal of Applied Psychology*, *60*(6), 720-729.

Beck, K. (2000). *Extreme programming explained: embrace change*. Reading: Addison-Wesley Longman Publishing Co., Inc.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., et al. (2001). Manifesto for Agile Software Development. Retrieved 25 October 2011, from http://www.agilemanifesto.org/

Bel, G., Fageda, X., & Warner, M. E. (2010). Is Private Production of Public Services Cheaper than Public Production? A meta-regression analysis of solid waste and water services. *Journal of Policy Analysis and Management, 29*(3), 553-577.

Belout, A., & Gauvreau, C. (2004). Factors influencing project success: the impact of human resource management. *International Journal of Project Management, 22*(1), 1-11.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly, 11*(3), 369-386.

Berg, B. L. (2009). *Qualitative research methods for the social sciences* (7th ed.). Boston: Allyn & Bacon.

Berlo, D. K. (1960). *The process of communication: an introduction to theory and practice*: Holt, Rinehart and Winston.

Betz, F. (2010). *Managing Science: Methodology and Organization of Research*: Springer.

Bezrukova, K., Jehn, K. A., Zanutto, E. L., & Thatcher, S. M. B. (2009). Do Workgroup Faultlines Help or Hurt? A Moderated Model of Faultlines, Team Identification, and Group Performance. *Organization Science, 20*(1), 35-50.

Bhatta, G. (2003). “Don’t just do something, stand there!”: Revisiting the Issue of Risks in Innovation in the Public Sector. *Innovation Journal*.

Bhopal, R. (2006). Race and ethnicity: responsible use from epidemiological and public health perspectives. *The Journal of Law, Medicine & Ethics, 34*(3), 500-507.

Bielefeldt, A. (2009). *Cognitive diversity and the performance of freshman engineering teams.* Paper presented at the ASEE Annual Conference and Exposition, Conference Proceedings.

Biemann, T., & Kearney, E. (2010). Size Does Matter: How Varying Group Sizes in a Sample Affect the Most Common Measures of Group Diversity. *Organizational Research Methods, 13*(3), 582-599.

Black, J., Hashimzade, N., & Myles, G. (Eds.). (2009) A Dictionary of Economics. Oxford: Oxford University Press.

Blau, P. M. (1964). *Exchange and power in social life*: Transaction Books.

Blau, P. M. (1977). *Inequality and heterogeneity: A primitive theory of social structure*: Free Press New York.

Boden, A., Nett, B., & Wulf, V. (2009). Operational and Strategic Learning in Global Software Development - Implications from two Offshoring Case Studies in Small Enterprises. *Software, IEEE, PP*(99), 1-1.

Boehm, B. W. (1981). Software engineering economics. *Englewood Cliffs*.

Boehm, B. W. (1987). Improving software productivity. *Computer, 20*(9), 43-57.

Boehm, B. W. (2005). *Some Future Trends and Implications for Systems and Software Engineering Processes*. Berlin Springer-Verlag.

Boehm, B. W. (2006). *A view of 20th and 21st century software engineering*. Paper presented at the 28th International Conference on Software Engineering.

Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., et al. (2000a). *COCOMO II Model Definition Manual, Version 2.1*. Los Angeles: Center for Software Engineering, University of Southern California.

Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., et al. (2000b). *Software Cost Estimation with COCOMO II*. New Jersey: Prentice Hall.

Boehm, B. W., Penedo, M., Stuckle, E., Williams, R., & Pyster, T. (2007). A software development environment for improving productivity. *Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research, 17*(6), 245.

Boehm, B. W., & Turner, R. (2005). *Balancing agility and discipline: A guide for the perplexed*. Reading: Addison-Wesley.

Bollen, K. A., & Davis, W. R. (2009). Causal Indicator Models: Identification, Estimation, and Testing. *Structural Equation Modeling: A Multidisciplinary Journal, 16*(3), 498 - 522.

Bonacin, R., Baranauskas, M. C. C., & Rodrigues, M. A. (2009). An Agile Process Model for Inclusive Software Development. In J. Filipe & J. Cordeiro (Eds.), *Enterprise Information Systems* (Vol. 24, pp. 807-818). Berlin Springer

Bond, M. H. (1988). Finding universal dimensions of individual variation in multicultural studies of values: The Rokeach and Chinese value surveys. *Journal of personality and social psychology, 55*(6), 1009-1015.

Boneau, C. A. (1960). The effects of violations of assumptions underlying the t test. *Psychological bulletin, 57*(1), 49.

Borchers, G. (2003, 3-10 May 2003). *The software engineering impacts of cultural factors on multi-cultural software development teams.* Paper presented at the 25th International Conference on Software Engineering, 2003.

Borland, J. H. (1990). Postpositivist Inquiry: Implications of the "New Philosophy of Science" for the Field of the Education of the Gifted. *Gifted Child Quarterly, 34*(4), 161-167.

Bostrom, R. P., Gupta, S., & Thomas, D. (2009). A Meta-Theory for Understanding Information Systems Within Sociotechnical Systems. *Journal of Management Information Systems, 26*(1), 17-47.

Boudreau, M.-C., Gefen, D., & Straub, D. W. (2001). Validation in Information Systems Research: A State-of-the-Art Assessment. *MIS Quarterly, 25*(1), 1-16.

Boyatzis, R. E. (1998). *Transforming qualitative information: thematic analysis and code development*: Sage Publications.

Boyatzis, R. E., & Sala, F. (2004). The Emotional Competence Inventory (ECI). In G. Geher (Ed.), *Measuring emotional intelligence: Common ground and controversy* (pp. 147–180). New York: Nova Science.

Bozeman, B., & Feeney, M. K. (2011). *Rules and Red Tape: A Prism for Public Administration Theory and Research*: M E Sharpe Inc.

Bozeman, B., & Kingsley, G. (1998). Risk culture in public and private organizations. *Public Administration Review, 58*(2), 109-118.

Brandes, L., Franck, E., & Theiler, P. (2009). The effect from national diversity on team production - expirical evidence from the sports industry. *Schmalenbach Business Review (SBR), 61*(2), 225-246.

Breitmayer, B. J., Ayres, L., & Knafl, K. A. (1993). Triangulation in Qualitative Research: Evaluation of Completeness and Confirmation Purposes. *Journal of Nursing Scholarship, 25*(3), 237-243.

Bretschneider, S. (1990). Management information systems in public and private organizations: An empirical test. *Public Administration Review*, 536-545.

Briand, L. C., El Emam, K., & Bomarius, F. (1998, 19-25 Apr 1998). *COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment.* Paper presented at the 1998 International Conference on Software Engineering.

Brodbeck, F. C. (2001). Communication and performance in software development projects. *European Journal of Work and Organizational Psychology, 10*(1), 73-94.

Brodbeck, F. C., Guillaume, Y. R. F., & Lee, N. J. (2011). Ethnic Diversity as a Multilevel Construct. *Journal of Cross-Cultural Psychology, 42*(7), 1198-1218.

Brooks, F. P. (1995). *The Mythical Man-Month: Essays on Software Engineering. 20th Anniversary ed.*: Addison-Wesley.

Brown, J. D. (2011). Likert items and scales of measurement? *Statistics, 15*(1).

Broy, M. (2006). *Challenges in automotive software engineering*. Paper presented at the Proceedings of the 28th international conference on Software engineering.

Bruce, A., & Langdon, K. (2000). *Project Management*: Dorling Kindersley.

Bryman, A. (2004). Triangulation. In M. Lewis-Beck, A. Bryman & T. Liao (Eds.), *The Sage encyclopedia of social science research methods* (pp. 1142-1143). Thousand Oaks: Sage Publications, Inc.

Bryman, A. (2006). Integrating quantitative and qualitative research: how is it done? *Qualitative Research, 6*(1), 97-113.

Bryman, A., Bresnen, M., Beardsworth, A., & Keil, T. (1988). Qualitative research and the study of leadership. *Human Relations, v41*(n1), p13(17).

Bureau of Labor Statistics. (2007). *Occupational Outlook Handbook, 2008-09 Edition*. Washington: United States Government.

Burgoon, M., Hunsaker, F. G., & Dawson, E. J. (1994). *Human communication*: Sage Publications.

Burningham, C., & West, M. A. (1995). Individual, Climate, and Group Interaction Processes as Predictors of Work Team Innovation. *Small Group Research, 26*(1), 106-117.

Burns, N., & Grove, S. (2005). *The practice of nursing research: Conduct, critique, and utilization*: WB Saunders Co.

Callister, P., Didham, R., & Kivi, A. (2009). Who are we?: The conceptualisation and expression of ethnicity. *Official Statistics Research Series, 4*.

Campbell, J., McDonald, C., & Sethibe, T. (2010). Public and private sector IT governance: Identifying contextual differences. *Australasian Journal of Information Systems, 16*(2).

Canalys. (2011). Google's Android becomes the world's leading smart phone platform. Retrieved 1 October 2012, from http://www.canalys.com/newsroom/google%E2%80%99s-android-becomes-world%E2%80%99s-leading-smart-phone-platform

Canas, K., & Sondak, H. (2008). *Opportunities and challenges of workplace diversity*. New Jersey: Prentice Hall.

Candrlic, S., Pavlic, M., & Poscic, P. (2006). *Model of a system for team software development.* Paper presented at the 28th International Conference on Information Technology Interfaces, 2006. .

Capezio, P. (1998). *Winning teams: making your team productive & successful*. New Jersey: Career Press.

Capretz, L. F., & Ahmed, F. (2010). Making Sense of Software Development and Personality Types. *IT Professional, 12*(1), 6-13.

Caracelli, V. J., & Greene, J. C. (1997). Crafting mixed-method evaluation designs. *New directions for evaluation, 1997*(74), 19-32.

Carayannis, E., Kaloudis, A., & Mariussen, Å. (2008). *Diversity in the Knowledge Economy and Society: Heterogeneity, Innovation and Entrepreneurship*. Northampton: Edward Elgar Publishing.

Carmel, E., & Abbott, P. (2007). Why 'nearshore' means that distance matters. *Commun. ACM, 50*(10), 40-46.

Carnegie Mellon Software Engineering Institute. (2006). *CMMI for Development, Version 1.2*. Pittsburgh: Author.

Caroline, S. (2001). Qualitative research requires quality concepts of its own. *Management Decision, 39*(7), 551-556.

Carter, K. N., Hayward, M., Blakely, T., & Shaw, C. (2009). How much and for whom does self-identified ethnicity change over time in new zealand? Results from a longitudinal study. *Social Policy Journal of New Zealand*(36), 32-45.

Cascio, W. F. (2003). *Managing human resources* (6th ed.). New York: McGraw-Hill Irwin.

Cassidy-MacKenzie, B. (2013a). *April 2013 Editorial*. Wellington: EEO Trust.

Cassidy-MacKenzie, B. (2013b). *EEO Update*. Wellington: EEO Trust.

Castells, M. (2009). *The rise of the network society*: Wiley-Blackwell.

Cerpa, N., & Verner, J. M. (2009). Why Did Your Project Fail? *Communications of the ACM, 52*(12), 130-134.

Chao, G. T., & Moon, H. (2005). The Cultural Mosaic: A Metatheory for Understanding the Complexity of Culture. *Journal of Applied Psychology, 90*(6), 1128-1140.

Charette, R. N. (2005). Why software fails [software failure]. *Spectrum, IEEE, 42*(9), 42-49.

Charette, R. N. (2009). This Car Runs on Code. http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research, 2*(6), 429-444.

Chatenier, E. D., Verstegen, J. A. A. M., Biemans, H. J. A., Mulder, M., & Omta, O. (2009). The challenges of collaborative knowledge creation in open innovation teams. *Human Resource Development Review, 8*(3), 350-381.

Chen, J., Sun, P., & McQueen, R. (2010). The impact of national cultures on structured knowledge transfer. *Journal of Knowledge Management, 14*(2), 228-242.

Chen, M. (2013). The Panel with Mai Chen and Graham Bell (Part 2). On *Afternoons*. Wellington: Radio New Zealand.

Cherry, S., & Robillard, P. N. (2009). *Audio-video recording of ad hoc software development team interactions.* Paper presented at the ICSE Workshop on Cooperative and Human Aspects on Software Engineering, 2009.

Chin, W. W., Marcolin, B. L., & Newsted, P. R. (2003). A partial least squares latent variable modeling approach for measuring interaction effects: results from a Monte Carlo simulation study and an electronic-mail emotion/adoption study. *Information Systems Research, 14*(2), 189(131).

Chirkov, V. I., Ryan, R. M., & Willness, C. (2005). Cultural Context and Psychological Needs in Canada and Brazil: Testing a Self-Determination Approach to the Internalization of Cultural Practices, Identity, and Well-Being. *Journal of Cross-Cultural Psychology, 36*(4), 423-443.

Choi, S., & Rainey, H. G. (2010). Managing Diversity in U.S. Federal Agencies: Effects of Diversity and Diversity Management on Employee Perceptions of Organizational Performance. *Public Administration Review, 70*(1), 109-121.

Chowdhury, S. (2005). Demographic diversity for building an effective entrepreneurial team: is it important? *Journal of Business Venturing, 20*(6), 727-746.

Chua, B. B., & Verner, J. (2010). Examining Requirements Change Rework Effort: A Study. *International Journal of Software Engineering & Applications, 1*(3), 48-64.

Chulani, S., Boehm, B., & Steece, B. (1999). Bayesian analysis of empirical software engineering cost models. *IEEE Transactions on Software Engineering, 25*(4), 573-583.

Clarke, N. (2010). Emotional intelligence and its relationship to transformational leadership and key project manager competences. *Project Management Journal, 41*(2), 5-20.

Clason, D., Dormody, T., & Scales, L. (1994). Analyzing data measured by individual Likert-type items. *Studies, 3*(7), 4.

Cleland, D., & Ireland, L. (2006). *Project Management: Strategic Design and Implementation*: McGraw-Hill Companies,Incorporated.

Cockburn, A. (2002). *Agile Software Development*. Indianapolis: Addison-Wesley.

Cockburn, A. (2007). *Agile software development: the cooperative game* (2 ed.). Reading: Addison-Wesley.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*: L. Erlbaum Associates.

Cohen, J. (1994). The earth is round (p<.05). *American Psychologist, 49*(12), 997-1003.

Congalton, J. (2011). *ISBSG Analysis Report: Government and Non-Government Software Project Performance*: International Software Benchmarking Standards Group.

Cook, C., & Churcher, N. (2006). *Constructing real-time collaborative software engineering tools using CAISE, an architecture for supporting tool development*. Paper presented at the Proceedings of the 29th Australasian Computer Science Conference - Volume 48.

Coolican, H. (2009). *Research methods and statistics in psychology* (5th ed.). London: Hodder & Stoughton Educational.

Cooper, W. W., Seiford, L. M., & Zhu, J. (2011). *Handbook on Data Envelopment Analysis*: Springer.

Cormack, D. (2010). *The politics and practice of counting: ethnicity in official statistics in Aotearoa/New Zealand*. Wellington: Te Rōpū Rangahau Hauora a Eru Pōmare.

Court, R. (2013, 4 May). Tackling workplace culture issues. *New Zeland Herald*. Retrieved from http://www.nzherald.co.nz/business/news/article.cfm?c_id=3&objectid=10881394

Craig, R. T. (1999). Communication theory as a field. *Communication theory, 9*(2), 119-161.

Cramton, C. D., & Webber, S. S. (2005). Relationships among geographic dispersion, team processes, and effectiveness in software development work teams. *Journal of Business Research, 58*(6), 758-765.

Creswell, J. (2009). *Research Design: Qualitative, Quantitative and Mixed Methods Approach* (3rd ed.). Los Angeles: Sage.

Creswell, J., & Clark, V. (2007). *Designing and conducting mixed methods research*. Los Angeles: Sage.

Creswell, J., & Clark, V. (2011). *Designing and Conducting Mixed Methods Research*: SAGE Publications.

Cropanzano, R., & Mitchell, M. S. (2005). Social Exchange Theory: An Interdisciplinary Review. *Journal of Management, 31*(6), 874-900.

Cugola, G., & Ghezzi, C. (1998). Software Processes: a Retrospective and a Path to the Future. *Software Process: Improvement and Practice, 4*(3), 101-123.

Cunningham, G. (2007). Perceptions As Reality: The Influence of Actual and Perceived Demographic Dissimilarity. *Journal of Business & Psychology, 22*(1), 79-89.

Curtis, E. F., & Dreachslin, J. L. (2008). Diversity management interventions and organizational performance: A synthesis of current literature. *Human Resource Development Review, 7*(1), 107-134.

Cusumano, M. A., Crandall, W., MacCormack, A., & Kemerer, C. F. (2009). Critical Decisions in Software Development: Updating the State of the Practice. *Software, IEEE, 26*(5), 84-87.

Dafoulas, G., & Macaulay, L. (2001). Investigating cultural differences in virtual software teams. *Electronic Journal on Information Systems in Developing Countries, 7*(4), 1-14.

Dahlin, K. B., Weingart, L. R., & Hinds, P. J. (2005). Team Diversity And Information Use. *Academy of Management Journal, 48*(6), 1107-1123.

Daim, T. U., Ha, A., Reutiman, S., Hughes, B., Pathak, U., Bynum, W., et al. (2012). Exploring the communication breakdown in global virtual teams. *International Journal of Project Management, 30*(2), 199-212.

Damian, D. E., & Zowghi, D. (2003, 6-9 Jan. 2003). *An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations.* Paper presented at the 36th Annual Hawaii International Conference on System Sciences, 2003.

Davis, N., & Mullaney, J. (2003). *The Team Software Process (TSP) in Practice: A Summary of Recent Results*. Pittsburgh: Software Engineering Institute, Carnegie Mellon.

Dawes, S. S., Pardo, T. A., Simon, S., Cresswell, A. M., LaVigne, M. F., Andersen, D. F., et al. (2004). *Making Smart IT Choices: Understanding Value and Risk in Government IT Investments*: Center for Technology in Government, State University of New York at Albany.

De Farias Jr, I. H., De Azevedo, R. R., De Moura, H. P., & Da Silva, D. S. M. (2012). *Elicitation of communication inherent risks in distributed software development*.

de Souza, C., Quirk, S., Trainer, E., & Redmiles, D. (2007). *Supporting collaborative software development through the visualization of socio-technical dependencies.* Paper presented at the 2007 International ACM Conference on Supporting Group Work, Sanibel Island, Florida, USA.

DeCecco, J. P., & Elia, J. P. (1993). A Critique and Synthesis of Biological Essentialism and Social Constructionist Views of Sexuality and Gender. *Journal of Homosexuality, 24*(3), 1 - 26.

DeMarco, T. (1995). *Why does software cost so much?: and other puzzles of the information age*: Dorset House Publishing Co., Inc. New York, NY, USA.

DeMarco, T., & Lister, T. R. (1999). *Peopleware: productive projects and teams*: Dorset House Pub.

Denzin, N. (1997). Triangulation in educational research *Educational research, methodology and measurement: An international handbook* (pp. 318-322).

Devnani-Chulani, S. (1999). *Bayesian analysis of software cost and quality models.* University Of Southern California.

Dibbern, J., Goles, T., Hirschheim, R., & Jayatilaka, B. (2004). Information systems outsourcing: a survey and analysis of the literature. *SIGMIS Database, 35*(4), 6-102.

Dong, F., Li, M., Li, J., Yang, Y., & Wang, Q. (2009). *Effect of staffing pattern on software project: An empirical analysis.* Paper presented at the 2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, Lake Buena Vista, FL.

Dubé, L., & Paré, G. (2003). Rigor in information systems positivist case research: current practices, trends, and recommendations. *MIS Quarterly, 27*(4), 597-635.

Dulewicz, V., & Higgs, M. (2005). Assessing leadership styles and organisational context. *Journal of Managerial Psychology, 20*(2), 105-123.

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

Earley, P. C., & Mosakowski, E. (2000). Creating hybrid team cultures: An empirical test of transnational team functioning. *Academy of Management Journal, 43*(1), 26-49.

Ebert, C., & Dumke, R. (2007). Measurement Repositories *Software Measurement* (pp. 471-486): Springer Berlin Heidelberg.

Edgell, S. E., & Noon, S. M. (1984). Effect of violation of normality on the t test of the correlation coefficient. *Psychological bulletin, 95*(3), 576-583.

EEO Trust. (2013). *Rise in workplace racism is unacceptable*. Wellington: EEO Trust.

Egan, R., W. , Tremaine, M., Fjermestad, J., Milewski, A., & O'Sullivan, P. (2006). *Cultural Differences in Temporal Perceptions and its Application to Running Efficient Global Software Teams.* Paper presented at the Internaltional Conference on Global Software Engineering, 2006. ICGSE '06.

Egan, R., W. , Tremaine, M., Fjermestad, J., Suling, Z., Milewski, A. E., & O'Sullivan, P. (2009, 13-16 July 2009). *How Urgent is Urgent? The Impact of Culturally-Based Temporal Perceptions* *on Virtual Teams.* Paper presented at the Fourth IEEE International Conference on Global Software Engineering.

Ehrlich, K., & Chang, K. (2006). *Leveraging expertise in global software teams: Going outside boundaries.* Paper presented at the International Conference on Global Software Engineering, 2006.

Ehsan, N., Waheed, K. Z., Asghar, U., Nawaz, M. T., Mirza, E., & Sarwar, S. Z. (2010). *Effects of project manager's competency on project success.* Paper presented at the IEEE International Conference on Management of Innovation and Technology (ICMIT).

Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: opportunities and challenges. *Academy of Management Journal, 50*(1), 25-32.

Ellis, T. (1995). *COTS integration in software solutions-A cost model*. Manassas: Loral Federal Systems.

Ely, R. J., & Thomas, D. A. (2001). Cultural Diversity at Work: The Effects of Diversity Perspectives on Work Group Processes and Outcomes. *Administrative Science Quarterly, 46*(2), 229-273.

Erdogmus, H. (2009). Diversity and Software Development. *IEEE Software, 26*(3), 2-4.

Eriksen, T. (2003). Ethnicity and Nationalism. *Journal of the Royal Anthropological Institute, 9*(4), 803-805.

Fagin, R., Kumar, R., & Sivakumar, D. (2003). Comparing Top k Lists. *Society for Industrial and Applied Mathematics, 17*(1).

Fairley, R. E. (2007). The influence of COCOMO on software engineering education and training. *Journal of Systems and Software, 80*(8), 1201-1208.

Fang, T. (2003). A critique of Hofstede's fifth national culture dimension. *International journal of cross cultural management, 3*(3), 347-368.

Fang, T., & Faure, G. O. (2011). Chinese communication characteristics: A Yin Yang perspective. *International Journal of Intercultural Relations, 35*(3), 320-333.

Faraj, S., & Sproull, L. (2000). Coordinating Expertise in Software Development Teams. *Management Science, 46*(12), 1554-1568.

Farrell, M. J. (1957). The Measurement of Productive Efficiency. *Journal of the Royal Statistical Society. Series A (General), 120*(3), 253-290.

Feeney, M. K. (2012). Organizational Red Tape: A Measurement Experiment. *Journal of Public Administration Research & Theory, 22*(3), 427-444.

Fernandez, S., Cho, Y. J., & Perry, J. L. (2010). Exploring the link between integrated leadership and public sector performance. *The Leadership Quarterly, 21*(2), 308-323.

Feuer, M. J., Towne, L., & Shavelson, R. J. (2002). Scientific Culture and Educational Research. *Educational Researcher, 31*(8), 4-14.

Fink, A. (2003). *The survey kit: How to sample in surveys*: Sage Publications, Inc.

Flood, R. L., & Jackson, M. C. (1991). *Creative Problem Solving: Total Systems Intervention*. New York: Wiley.

Ford, C., & Harawa, N. T. (2010). A new conceptualization of ethnicity for social epidemiologic and health equity research. *Social science & medicine (1982), 71*(2), 251.

Foulds, L., Quaddus, M., & West, M. (2007, 11-13 July 2007). *Structural Equation Modelling of Large-scale Information System Application Development Productivity: the Hong Kong Experience.* Paper presented at the 6th IEEE/ACIS International Conference on Computer and Information Science, 2007.

Foulds, L., & West, M. (2007). The productivity of large business information system development. *International Journal of Business Information Systems, 2*(2), 162-181.

Fowler, M. (2005). The New Methodology. Retrieved 28 April 2010, from http://martinfowler.com/articles/newMethodology.html

Frame, J. D. (2003). *Managing Projects in Organizations: How to Make the Best Use of Time, Techniques, and People*: Wiley.

Franke, U., Narman, P., Hook, D., & Lillieskold, J. (2010, 18-22 July 2010). *Factors affecting successful project management of technology-intense projects.* Paper presented at the Technology Management for Global Economic Growth (PICMET), 2010 Proceedings of PICMET '10:.

Fraser, S. (2009). Software "Best" Practices: Agile Deconstructed. In M. O. Frank Bomarius, Päivi Jaring and Pekka Abrahamsson (Ed.), *Product-Focused Software Process Improvement* (Vol. 32, pp. 8-13). Berlin Springer

Fraser, S., Boehm, B., Brooks Jr, F., DeMarco, T., Lister, T., Rising, L., et al. (2007, 20-26 May 2007). *Retrospectives on Peopleware.* Paper presented at the 29th International Conference on Software Engineering - Companion, 2007.

Fried, L. (1991). Team size and productivity in systems development. *Information Systems Management, 8*(3), 27.

Frink, D. D., Robinson, R. K., Reithel, B., Arthur, M. M., Ammeter, A. P., Ferris, G. R., et al. (2003). Gender Demography and Organization Performance: A Two-Study Investigation With Convergence. *Group & Organization Management, 28*(1), 127-147.

Fuchs, A. (2013). *Nonlinear Dynamics in Complex Systems*: Springer.

Fuks, H., Raposo, A. B., Gerosa, M. A., & Lucena, C. J. P. (2005). Applying the 3C model to groupware development. *International Journal of Cooperative Information Systems, 14*(2/3), 299-328.

Gable, G. G. (1994). Integrating case study and survey research methods: an example in information systems. *European Journal of Information Systems, 3*(2), 112-126.

Galorath. (2012). *SEER for Software: Product Brief*. Andover: Galorath.

Gannon, J. D. (1979). Human Factors in Software Engineering. *Computer, 12*(12), 6-7.

Ganssle, J. G. (2008). *The Art of Designing Embedded Systems*: Elsevier / Newnes.

Gauld, R. (2007). Public sector information system project failures: Lessons from a New Zealand hospital organization. *Government Information Quarterly, 24*(1), 102-114.

Gaur, A. S., & Gaur, S. S. (2009). *Statistical Methods for Practice and Research: A Guide to Data Analysis Using SPSS*: Sage Publications.

Gehring, D. R. (2007). Applying traits theory of leadership to project management. *Project Management Journal, 38*(1), 44-54.

Ghezzi, C., Jazayeri, M., & Mandrioli, D. (2004). Software Qualities and Principles. Chapter 101 of Allen Tucker. Computer Science Handbook: Chapman & Hall/CRC.

Gibson, D. L., Goldenson, D. R., & Kost, K. (2006). *Performance Results of CMMI-Based Process Improvement*. Pittsburgh Carnegie Mellon Software Engineering Institute.

Gillard, S., & Johansen, J. (2004). Project management communication: a systems approach. *Journal of Information Science, 30*(1), 23.

Glover, J., & Evans, C. (2011). Women's representation in ICT employment: Approaches to bringing about organizational change. *International Journal of Gender, Science and Technology, 6*(3), 45-55.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The qualitative report, 8*(4), 597-607.

Goodwin, R. (1999). *Personal relationships across cultures*: Routledge.

Gordon, R., Hale, K., Hardcastle, J., Graham, C., Kjeldsen, P., Shiffler, G., et al. (April 2012). *Forecast Alert: IT Spending, Worldwide, 1Q12 Update*. Stamford: Gartner.

Goyal, S., Maruping, L., & Robert, L. (2008). Diversity and conflict in teams: a faultline model perspective. *Academy of Management Proceedings*, 1-6.

Green, S. B., & Salkind, N. J. (2011). *Using SPSS for Windows and Macintosh: Analyzing and Understanding Data: International Edition*: Pearson Education, Limited.

Greene, J. C. (2007). *Mixed methods in social inquiry*: John Wiley & Sons.

Greene, J. C., Caracelli, V., & Graham, W. (1989). Toward a conceptual framework for mixed-method evaluation designs. *Educational evaluation and policy analysis, 11*(3), 255.

Greene, J. C., & Caracelli, V. J. (1997). Defining and describing the paradigm issue in mixed-method evaluation. *New directions for evaluation, 1997*(74), 5-17.

Griffiths, D. (1980). A Pragmatic Approach to Spearman's Rank Correlation Coefficient. *Teaching Statistics, 2*(1), 10-13.

Gujarati, D. (1970). Use of dummy variables in testing for equality between sets of coefficients in two linear regressions: a note. *The American Statistician, 24*(1), 50-52.

Gumperz, J. J. (1982). *Discourse strategies*: Cambridge University Press.

Guzzo, R. A., & Dickson, M. W. (1996). Teams in organizations: Recent research on performance and effectiveness. *Annual Review of Psychology, 47*(1), 307.

Haaland, K., Stamelos, I., Ghosh, R., & Glott, R. (2009). *On the Approximation of the Substitution Costs for Free/Libre Open Source Software*. Paper presented at the Fourth Balkan Conference inInformatics, 2009.

Hackney, R., Jones, S., & Losch, A. (2007). Towards an e-Government efficiency agenda: the impact of information and communication behaviour on e-Reverse auctions in public sector procurement. *Eur J Inf Syst, 16*(2), 178-191.

Haken, H. (2006). *Information and Self-organization: A Macroscopic Approach to Complex Systems*. Stuttgart: Physica-Verlag.

Hall, E. T. (1983). *The dance of life: The other dimension of time*. New York: Anchor Press/Doubleday.

Hall, T., Wilson, D., Rainer, A., & Jagielska, D. (2007). *Communication: the neglected technical skill?* Paper presented at the Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce.

Hall, W. A., & Callery, P. (2001). Enhancing the Rigor of Grounded Theory: Incorporating Reflexivity and Relationality. *Qualitative Health Research, 11*(2), 257-272.

Halstead, M. H. (1977). *Elements of software science*: Elsevier.

Hambrick, D. C., Cho, T. S., & Chen, M.-J. (1996). The Influence of Top Management Team Heterogeneity on Firms' Competitive Moves. *Administrative Science Quarterly, 41*(4), 659-684.

Harbich, T., & Alisch, K. (2007). *Accuracy of Estimation Models with Discrete Parameter Values shown on the Example of COCOMO II*. Paper presented at the Proceedings of Software Measurement European Forum (SMEF). Rome.

Harding, J. F., Sibley, C. G., & Robertson, A. (2011). New Zealand= Māori, New Zealand= bicultural: Ethnic group differences in a national sample of Māori and Europeans. *Social Indicators Research, 100*(1), 137-148.

Harris-Kojetin, L. D., Fowler Jr, F. J., Brown, J. A., Schnaier, J. A., & Sweeny, S. F. (1999). The Use of Cognitive Testing to Develop and Evaluate CAHPS (TM) 1.0 Core Survey Items. *Medical Care, 37*(3), MS10.

Harrison, D. A., & Klein, K. J. (2007). What's the difference? Diversity constructs as separation, variety, or disparity in organizations. *Academy of Management Review, 32*(4), 1199-1228.

Harrits, G. S. (2011). More Than Method?: A Discussion of Paradigm Differences Within Mixed Methods Research. *Journal of Mixed Methods Research, 5*(2), 150-166.

Hartley, S. (2008). *Project Management*: Pearson Education Australia.

Hayes, J. H. (2003). Do you like Pina Coladas? How improved communication can improve software quality. *Software, IEEE, 20*(1), 90-92.

Healy, M., & Perry, C. (2000). Comprehensive criteria to judge validity and reliability of qualitative research within the realism paradigm. *Qualitative Market Research: An International Journal, 3*(3), 118-126.

Heemstra, F. J., Kusters, R. J., & de Man, H. (2003, 30 Sept.-1 Oct. 2003). *Guidelines for managing bias in project risk management.* Paper presented at the International Symposium on Empirical Software Engineering, 2003.

Heijstek, W., & Chaudron, M. R. V. (2009). *Empirical Investigations of Model Size, Complexity and Effort in a Large Scale, Distributed Model Driven Development Process.* Paper presented at the 35th Euromicro Conference on Software Engineering and Advanced Applications, 2009.

Henson, R. K., Hull, D. M., & Williams, C. S. (2010). Methodology in Our Education Research Culture. *Educational Researcher, 39*(3), 229-240.

Herbsleb, J. (2007). *Global Software Engineering: The Future of Socio-technical Coordination*. Paper presented at the 2007 Future of Software Engineering.

Herbsleb, J., & Mockus, A. (2003). An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering, 29*(6), 481-494.

Herbsleb, J., & Moitra, D. (2001). Global software development. *IEEE Software, 18*(2), 16-20.

Highsmith, J. (2002). *Agile software development ecosystems*. Reading: Addison-Wesley Longman Publishing Co., Inc.

Hill, P. (2010). *Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration*: McGraw-Hill.

Hill, R. S. (2010). Fitting Multiculturalism into Biculturalism: Maori-Pasifika Relations in New Zealand from the 1960s. *Ethnohistory, 57*(2), 291-319.

Hill, T., & Lewicki, P. (2006). *Statistics: methods and applications : a comprehensive reference for science, industry, and data mining*: StatSoft.

Hofmann, H., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software*, 58-66.

Hofstede, G. H. (1980a). Culture and organizations. *International Studies of Management & Organization*, 15-41.

Hofstede, G. H. (1980b). Motivation, leadership, and organization: Do American theories apply abroad. *Organizational Dynamics, 9*(1), 42-63.

Hofstede, G. H. (2006). What did GLOBE really measure? Researchers' minds versus respondents' minds. *Journal of International Business Studies, 37*(6), 882-896.

Hofstede, G. H., Hofstede, G. J., & Minkov, M. (2010). *Cultures and Organizations: Software of the Mind, Third Edition*: McGraw-Hill Education.

Hofstede, G. H., Neuijen, B., Ohayv, D. D., & Sanders, G. (1990). Measuring Organizational Cultures: A Qualitative and Quantitative Study across Twenty Cases. *Administrative Science Quarterly, 35*(2), 286-316.

Holmes, J., & Marra, M. (2010). Leadership and managing conflict in meetings. *Pragmatics, 14*(4).

Hoque, Z. (2006). *Methodological issues in accounting research: theories, methods and issues*: Spiramus Pr.

Horii, T., Jin, Y., & Levitt, R. (2005). Modeling and Analyzing Cultural Influences on Project Team Performance. *Computational & Mathematical Organization Theory, 10*(4), 305-321.

Horwitz, S. K. (2005). The Compositional Impact of Team Diversity on Performance: Theoretical Considerations. *Human Resource Development Review, 4*(2), 219-245.

Horwitz, S. K., & Horwitz, I. B. (2007). The effects of team diversity on team outcomes: A meta-analytic review of team demography. *Journal of Management, 33*(6), 987-1015.

House, R., Javidan, M., Hanges, P., & Dorfman, P. (2002). Understanding cultures and implicit leadership theories across the globe: an introduction to project GLOBE. *Journal of World Business, 37*(1), 3-10.

Hoyle, R., Harris, M., & Judd, C. (2002). *Research methods in social relations*. Kentucky: Wadsworth.

Hsu, J. S.-C., Shih, S.-P., Chiang, J. C., & Liu, J. Y.-C. (2011). The impact of transactive memory systems on IS development teams' coordination, communication, and performance. *International Journal of Project Management, 30*(3), 329-340.

Huang, X. S., Capretz, L. F., Ren, J., & Ho, D. (2003). *A neuro-fuzzy model for software cost estimation.* Paper presented at the Third International Conference on Quality Software, Dallas.

Hubbard, R., & Lindsay, R. M. (2008). Why P Values Are Not a Useful Measure of Evidence in Statistical Significance Testing. *Theory & Psychology, 18*(1), 69-88.

Hull, R., Gibbon, J., & Branzei, O. (2011). *The Third Sector*: Emerald Group Pub Limited.

Hultsch, D. F., MacDonald, S. W. S., Hunter, M. A., Maitland, S. B., & Dixon, R. A. (2002). Sampling and generalisability in developmental research: Comparison of random and convenience samples of older adults. *International Journal of Behavioral Development, 26*(4), 345-359.

Humphrey, W. S. (2000). *The Team Software ProcessSM (TSPSM)*. Pittsburgh: Software Engineering Institute, Carnegie Mellon.

Hunley, K. L., Healy, M. E., & Long, J. C. (2009). The global pattern of gene identity variation reveals a history of long-range migrations, bottlenecks, and local mate exchange: implications for biological race. *American journal of physical anthropology, 139*(1), 35-46.

Idri, A., Mbarki, S., & Abran, A. (2004). *Validating and understanding software cost estimation models based on neural networks.* Paper presented at the International Conference on Information and Communication Technologies: From Theory to Applications, 2004.

Inglehart, R. (2006). Mapping global values. *International studies in sociology and social anthropology, 104*, 11.

International Function Point Users Group. (2005). *Function Point Counting Practices Manual (Release 4.2.1)*. New Jersey Author.

International Function Point Users Group. (2010). *Function Point Counting Practices Manual (Release 4.3.1)*. New Jersey Author.

International Labour Organisation. (2008). *Skills for improved productivity, employment growth and development*. Geneva: International Labour Office.

International Software Benchmarking Standards Group. (2007). Guidelines for use of the ISBSG data.

International Software Benchmarking Standards Group. (2009). ISBSG Data Repository. Release 11.

Investment New Zealand. (2007). *New Zealand Software Development Industry*: New Zealand Trade and Enterprise.

ISO/IEC. (2008). ISO/IEC/IEEE Standard for Systems and Software Engineering - Software Life Cycle Processes, *IEEE STD 12207-2008* (pp. c1-138).

Jackson, S. E., Joshi, A., & Erhardt, N. L. (2003). Recent Research on Team and Organizational Diversity: SWOT Analysis and Implications. *Journal of Management, 29*(6), 801-830.

Jakobsson, U. (2004). Statistical presentation and analysis of ordinal data in nursing research. *Scandinavian journal of caring sciences, 18*(4), 437-440.

Jamieson, S. (2004). Likert scales: how to (ab) use them. *Medical Education, 38*(12), 1217-1218.

Janz, B. D., & Prasarnphanich, P. (2009). Freedom to Cooperate: Gaining Clarity Into Knowledge Integration in Information Systems Development Teams. *Ieee Transactions on Engineering Management*, *56*(4), 621-635.

Javidan, M., House, R. J., Dorfman, P. W., Hanges, P. J., & Luque, M. S. d. (2006). Conceptualizing and Measuring Cultures and Their Consequences: A Comparative Review of GLOBE's and Hofstede's Approaches. *Journal of International Business Studies, 37*(6), 897-914.

Jehn, K. A. (1995). A Multimethod Examination of the Benefits and Detriments of Intragroup Conflict. *Administrative Science Quarterly*, *40*(2), 256-282.

Jehn, K. A., & Bezrukova, K. (2004). A Field Study of Group Diversity, Workgroup Context, and Performance. *Journal of Organizational Behavior*, *25*(6), 703-729.

Jehn, K. A., Northcraft, G. B., & Neale, M. A. (1999). Why Differences Make a Difference: A Field Study of Diversity, Conflict, and Performance in Workgroups. *Administrative Science Quarterly*, *44*(4), 741-763.

Jiang, Z., Jiang, B., & Naudé, P. (2007). The effects of software size on development effort and software quality. *International Journal of Computer and Information Science and Engineering*, *1*(4), 230-234.

Jick, T. D. (1979). Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Administrative Science Quarterly, 24*(4), 602-611.

Johnson, R. B., & Onwuegbuzie, A. J. (2004). Mixed Methods Research: A Research Paradigm Whose Time Has Come. *Educational Researcher, 33*(7), 14-26.

Jones, C. (1995). Backfiring: converting lines of code to function points. *Computer*, *28*(11), 87-88.

Jones, C. (2000). *Software Assessments, Benchmarks, and Best Practices*. New Jersey: Addison-Wesley.

Jones, C. (2008). *Applied software measurement* (3rd ed.): McGraw-Hill Osborne Media.

Jones, M. L. (2007). *Hofstede-culturally questionable?* Paper presented at the Oxford Business & Economics Conference, Oxford.

Jorgensen, M., & Grimstad, S. (2012). Software Development Estimation Biases: The Role of Interdependence. *Software Engineering, IEEE Transactions on, 38*(3), 677-693.

Joshi, A., & Roh, H. (2009). The role of context in work team diversity research: A meta-analytic review. *Academy of Management Journal, 52*(3), 599-627.

Jugdev, K., & Muller, R. (2005). A retrospective look at our evolving understanding of project success. *Project Management Journal, 36*(4), 19-31.

Kakola, T. (2008). *Best Practices for International eSourcing of Software Products and Services.* Paper presented at the 41st Annual Hawaii International Conference on System Sciences.

Kangilaski, T. (2009, 3-5 Nov. 2009). *Asset Management software implementation challenges for electricity companies.* Paper presented at the Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE.

Karner, G. (1993). Resource estimation for objectory projects. *Objective Systems SF AB, 17*.

Karpowich, M., Sanders, T., & Verge, R. (1993). An economic analysis model for determining the custom versus commercial software tradeoffs. *Analytical methods in software engineering economics',(Springer-Verlag, 1993)*, 237-252.

Katzenbach, J. R., & Smith, D. K. (1993). The Discipline of Teams. *Harvard Business Review, 71*, 111-131.

Kirk, J., & Miller, M. (1986). *Reliability and validity in qualitative research*. Thousand Oaks: Sage Publications, Inc.

Kirkman, B. L., & Shapiro, D. L. (2001). The Impact of Team Members' Cultural Values on Productivity, Cooperation, and Empowerment in Self-Managing Work Teams. *Journal of Cross-Cultural Psychology, 32*(5), 597-617.

Kirsch, L. J., & Beath, C. M. (1996). The enactments and consequences of token, shared, and compliant participation in information systems development. *Accounting, Management and Information Technologies, 6*(4), 221-254.

Kitchenham, B. (1997). The Problem with Function Points. *Software, IEEE, 14*(2), 29-31.

Kitchenham, B. (2002). The question of scale economies in software--why cannot researchers agree? *Information and Software Technology, 44*(1), 13-24.

Kitchenham, B., Mendes, E., & Travassos, G. H. (2007). Cross versus Within-Company Cost Estimation Studies: A Systematic Review. *Software Engineering, IEEE Transactions on, 33*(5), 316-329.

Klein, C., DiazGranados, D., Salas, E., Le, H., Burke, C. S., Lyons, R., et al. (2009). Does Team Building Work? *Small Group Research, 40*(2), 181-222.

Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly, 23*(1), 67-93.

Klein, M., Sayama, H., Faratin, P., & Bar-Yam, Y. (2001). *What complex systems research can teach us about collaborative design.* Paper presented at the Sixth International Conference on Computer Supported Cooperative Work in Design, 2001.

Knutson, T. J., Komolsevin, R., Chatiketu, P., & Smith, V. R. (2003). A cross-cultural comparison of Thai and US American rhetorical sensitivity: implications for intercultural communication effectiveness. *International Journal of Intercultural Relations, 27*(1), 63-78.

Kochan, T., Bezrukova, K., Ely, R., Jackson, S., Joshi, A., Jehn, K., et al. (2003). The effects of diversity on business performance: report of the diversity research network. *Human Resource Management, 42*(1), 3-21.

Korn / Ferry Institute. (2013). *The Diversity & Inclusion Study*. Los Angeles: Author.

Krueger, R., & Casey, M. (2009). *Focus groups: A practical guide for applied research*. Thousand Oaks: Sage.

Kudyba, S., & Diwan, R. (2002). *Information technology, corporate productivity, and the new economy*: Greenwood Publishing Group.

Kukutai, T. H. (2007). White Mothers, Brown Children: Ethnic Identification of Maori-European Children in New Zealand. *Journal of Marriage and Family, 69*(5), 1150-1161.

Kundi, M. (2006). Causality and the Interpretation of Epidemiologic Evidence. *Environmental Health Perspectives, 114*(7), 969-974.

Kvale, S. (1996). *Interviews: an introduction to qualitative research interviewing*: Sage Publications.

Kvale, S. (2007). *Introduction to interview research. Doing Interviews. SAGE Publications, Ltd*. London, UK: SAGE Publications, Ltd.

Larman, C. (2004). *Agile and Iterative Development: A Manager's Guide*: Addison-Wesley.

Lau, D. C., & Murnighan, J. K. (1998). Demographic diversity and faultlines: The compositional dynamics of organizational groups. *Academy of Management Review*, 325-340.

LaVeist, T. A. (1994). Beyond dummy variables and sample selection: what health services researchers ought to know about race as a variable. *Health services research, 29*(1), 1.

Lazear, E. P. (1999). Culture and Language. *Journal of Political Economy, 107*(S6), S95-S126.

Lazear, E. P., & Shaw, K. L. (2007). Personnel Economics: The Economist's View of Human Resources. *Journal of Economic Perspectives, 21*(4), 91-114.

Lee, A. S., & Baskerville, R. L. (2003). Generalizing Generalizability in Information Systems Research. *Information Systems Research, 14*(3), 221-243.

Lee, A. S., & Hubona, G. S. (2009). A scientific basis for rigor in information systems research. *MIS Quarterly, 33*(2), 237-262.

Lee, A. S., & Nickerson, J. V. (2010). *Theory As a Case of Design: Lessons for Design from the Philosophy of Science*. Paper presented at the 2010 43rd Hawaii International Conference on System Sciences (HICSS).

Lee, G., & Xia, W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly, 34*(1), 87-114.

Lee, J. A. (2000). Adapting Triandis's Model of Subjective Culture and Social Behavior Relations to Consumer Behavior. *Journal of Consumer Psychology, 9*(2), 117-126.

Leech, N., & Onwuegbuzie, A. (2009). A typology of mixed methods research designs. *Quality & Quantity, 43*(2), 265-275.

Leon, A. (2010). How many lines of code does it take to create the Android OS? Retrieved 1 October 2012, from http://www.gubatron.com/blog/2010/05/23/how-many-lines-of-code-does-it-take-to-create-the-android-os/

Leong, C.-H., & Ward, C. (2000). Identity conflict in sojourners. *International Journal of Intercultural Relations, 24*(6), 763-776.

Leung, K., Bhagat, R. S., Buchan, N. R., Erez, M., & Gibson, C. B. (2005). Culture and International Business: Recent Advances and Their Implications for Future Research. *Journal of International Business Studies, 36*(4), 357-378.

Lewin, J., Meares, C., Cain, T., Spoonley, P., Pease, R., & Ho, E. (2011). *Namaste New Zealand: Indian Employers and Employers in Auckland* (No. 5): Massey University / University of Waikato.

Lewis, M. W., & Grimes, A. I. (1999). Metatriangulation: building theory from multiple paradigms. *Academy of Management Review, 24*(4), 672-690.

Lewis, T. L., & Smith, W. J. (2008). *Building software engineering teams that work: The impact of dominance on group conflict and Performance outcomes.* Paper presented at the 38th Annual Conference on Frontiers in Education.

Li, F., & Aksoy, L. (2007). Dimensionality of Individualism-Collectivism and Measurement Equivalence of Triandis and Gelfand's Scale. *Journal of Business and Psychology, 21*(3), 313-329.

Liang, T. P., Jiang, J., Klein, G. S., & Liu, J. Y. C. (2009). Software Quality as Influenced by Informational Diversity, Task Conflict, and Learning in Project Teams. *IEEE Transactions on Engineering Management, PP*(99), 1-11.

Liang, T. P., Jiang, J., Klein, G. S., & Liu, J. Y. C. (2010). Software Quality as Influenced by Informational Diversity, Task Conflict, and Learning in Project Teams. *IEEE Transactions on Engineering Management, 57*(3), 477-487.

Liang, T. P., Liu, C. C., Lin, T. M., & Lin, B. (2007). Effect of team diversity on software project performance. *Industrial Management And Data Systems, 107*(5), 636.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology, 22*(140), 55.

Lincoln, Y., & Guba, E. (2000). Paradigmatic controversies, contradictions, and emerging confluences *Handbook of qualitative research* (Vol. 2, pp. 163-188): Sage.

Lincoln, Y., Lynham, S., & Guba, E. (2011). Paradigmatic controversies, contradictions, and emerging confluences, revisited. In N. K. Denzin & Y. Lincoln (Eds.), *The Sage handbook of qualitative research* (pp. 97-128). Thousand Oaks: Sage.

Liu, Q., & Mintram, R. (2005). Preliminary data analysis methods in software estimation. *Software Quality Journal, 13*(1), 91-115.

Liu, Y., Luo, M., & Wei, X. (2008). *The effects of cultural diversity, conflict and conflict management on performance in global virtual teams.* Paper presented at the 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008.

Loden, M., & Rosener, J. (1991). *Workforce America!: Managing employee diversity as a vital resource*. Homewood: Business One Irwin.

Lokan, C. (1999). *An empirical study of the correlations between function point elements [software metrics]*. Paper presented at the Sixth International Software Metrics Symposium, 1999.

Lokan, C., & Mendes, E. (2006). *Cross-company and single-company effort models using the ISBSG database: a further replicated study*. Paper presented at the Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering.

López-Martín, C., Yáñez-Márquez, C., & Gutiérrez-Tornés, A. (2008). Predictive accuracy comparison of fuzzy models for software development effort of small programs. *Journal of Systems and Software, 81*(6), 949-960.

Ludwig Von, B. (1972). The History and Status of General Systems Theory. *The Academy of Management Journal*, *15*(4), 407-426.

Lunenburg, F., & Irby, B. (2007). *Writing a successful thesis or dissertation: tips and strategies for students in the social and behavioral sciences*. Thousand Oaks: Corwin Press.

MacDonell, S. G., Gray, A. R., & Calvert, J. M. (1999). *FULSOME: fuzzy logic for software metric practitioners and researchers.* Paper presented at the 6th International Conference on Neural Information Processing, 1999.

Mach, M., Dolan, S., & Tzafrir, S. (2010). The differential effect of team members' trust on team performance: The mediation role of team cohesion. *Journal of Occupational and Organizational Psychology, 83*(3), 771-794.

Mackie, D. M., Worth, L. T., & Asuncion, A. G. (1990). Processing of persuasive in-group messages. *Journal of personality and social psychology, 58*(5), 812-822.

Mahmood, M. A., Pettingell, K. J., & Shaskevich, A. I. (1996). Measuring Productivity of Software Projects: A Data Envelopment Analysis Approach. *Decision Sciences, 27*(1), 57-80.

Malhotra, N., & Birks, D. (2007). *Marketing Research: an applied approach: 3rd European Edition*: Pearson Education.

Mannix, E., & Neale, M. (2005). What differences make a difference? *Psychological Science in the Public Interest, 6*(2), 31.

Marban, O., Menasalvas, E., & Fernandez-Balzan, C. (2008). A cost model to estimate the effort of data mining projects (DMCoMo). *Information Systems, 33*(1), 133-150.

Marcoulides, G. A., & Saunders, C. (2006). PLS: A Silver Bullet? *MIS Quarterly,* pp. 211-211. Retrieved from http://ezproxy.massey.ac.nz/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,cookie,url,uid&db=heh&AN=21145593&site=ehost-live

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A Design Theory for Systems That Support Emergent Knowledge Processes. *MIS Quarterly, 26*(3), 179-212.

Marshall, M. (1996). The key informant technique. *Family Practice, 13*(1), 92-97.

Martin, R. C. (2003). *Agile Software Development: Principles, Patterns, and Practices*. Upper Saddle River: Prentice Hall PTR.

Martinez, C. (2008, 20-24 July 2008). *Intelligent real-time tools and visualizations for wide-area electrical grid reliability management.* Paper presented at the Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE.

Marttiin, P., Lehto, J. A., & Nyman, G. (2002). *Understanding and evaluating collaborative work in multi-site software projects - a framework proposal and preliminary results.* Paper presented at the 35th Annual Hawaii International Conference on System Sciences, 2002.

Mason, J. (2004). Semistructured Interview. In M. Lewis-Beck, A. Bryman & T. Liao (Eds.), *The Sage encyclopedia of social science research methods* (pp. 1020-1021). Thousand Oaks: Sage Publications, Inc.

Massey University. (2010a). Code of ethical conduct for research, teaching and evaluations involving human participants. Palmerston North: Author.

Massey University. (2010b). Human Ethics - Approval and notification procedures. Retrieved 27 May 2010, from http://www.massey.ac.nz/massey/research/research-ethics/human-ethics/approval.cfm

Mathieu, C. (2009). *Cultural diversity of distributed teams.* Paper presented at the SIGMIS CPR'09 - 2009 ACM SIGMIS Computer Personnel Research Conference.

Mathison, S. (1988). Why triangulate? *Educational Researcher*, *17*(2), 13.

Maxwell, J. A. (1992). Understanding and validity in qualitative research. *Harvard Educational Review*, *62*(3), 279.

Maxwell, J. A. (2004). Using Qualitative Methods for Causal Explanation. *Field Methods*, *16*(3), 243-264.

Maxwell, K. (2001). Collecting data for comparability: benchmarking software development productivity. *IEEE Software*, 22-25.

Maxwell, K., & Forselius, P. (2000). Benchmarking software development productivity. *Ieee Software, 17*(1), 80-88.

Maxwell, K., Van Wassenhove, L., & Dutta, S. (1996). Software development productivity of european space, military, and industrial applications. *Ieee Transactions on Software Engineering, 22*(10), 706-718.

McAtavey, J., & Nikolovska, I. (2010). Team collectivist culture: A remedy for creating team effectiveness. *Human Resource Development Quarterly, 21*(3), 307-316.

McCreery, J., & Moranta, V. (2009, 2-6 Aug. 2009). *Mapping team collaboration in software development projects.* Paper presented at the Portland International Conference on Management of Engineering & Technology, 2009.

McGuire, E. G. (1996). *Initial effects of software process improvement on an experienced software development team.* Paper presented at the Twenty-Ninth Hawaii International Conference on System Sciences, 1996.

Mendes, E., & Lokan, C. (2008). Replicating studies on cross- vs single-company effort models using the ISBSG Database. *Empirical Software Engineering, 13*(1), 3-37.

Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R., & Jazayeri, M. (2005). *Challenges in software evolution.* Paper presented at the Eighth International Workshop on Principles of Software Evolution.

Meredith, J. R., & Mantel, S. J. (2008). *Project Management: A Managerial Approach*: John Wiley & Sons.

Meredith, J. R., & Mantel, S. J. (2011). *Project Management: A Managerial Approach*: Wiley.

Meyer, B., Shemla, M., & Schermuly, C. C. (2011). Social category salience moderates the effect of diversity faultlines on information elaboration. *Small Group Research, 42*(3), 257.

Meynhardt, T., & Diefenbach, F. E. (2012). What Drives Entrepreneurial Orientation in the Public Sector? Evidence from Germany's Federal Labor Agency. *Journal of Public Administration Research & Theory, 22*(4), 761-792.

Miller, K. (2004). *Communication theories: perspectives, processes, and contexts*: McGraw-Hill.

Mingers, J. (2001). Combining IS research methods: towards a pluralist methodology. *Information Systems Research, 12*(3), 240-259.

Minkiewicz, A. (2008). *Estimating Software from Requirements*. Mt. Laurel: PRICE Systems.

Minkov, M., & Hofstede, G. (2011). The evolution of Hofstede's doctrine. *Cross Cultural Management: An International Journal, 18*(1), 10-20.

Mishra, S., Hazra, K., & Mall, R. (2011). A Survey of Metrics for Software Development Effort Estimation. *International Journal of Research and Reviews in Computer Science (IJRRCS), 2*(5).

Mittal, A., Parkash, K., & Mittal, H. (2010). Software cost estimation using fuzzy logic. *SIGSOFT Softw. Eng. Notes, 35*(1), 1-7.

Miyazaki, A. D., & Taylor, K. A. (2008). Researcher Interaction Biases and Business Ethics Research: Respondent Reactions to Researcher Characteristics. *Journal of Business Ethics, 81*(4), 779-795.

Modaff, D. P., DeWine, S., & Butler, J. A. (2008). *Organizational Communication: Foundations, Challenges, and Misunderstandings* (2nd ed.): Allyn & Bacon, Incorporated.

Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology, 52*(5), 480-491.

Molm, L. D. (2003). Theoretical Comparisons of Forms of Exchange. *Sociological Theory*, *21*(1), 1-17.

Montangero, C. (1999). The Software Process: Modelling and Technology. In J. Dernaime, B. Kaba & D. Wastell (Eds.), *Software Process: Principles, Methodology and Technology* (pp. 1-13). Berlin: Springer.

Moon, B. M., Hoffman, R. R., & Novak, J. (2011). *Applied Concept Mapping: Capturing, Analyzing, and Organizing Knowledge*: CRC Press.

Morgan, D. (2007). Paradigms lost and pragmatism regained. *Journal of Mixed Methods Research, 1*(1), 48.

Morgan, G. (2006). *Images of Organization*. Thousand Oaks: Sage.

Morse, J. M., Barrett, M., Mayan, M., Olson, K., & Spiers, J. (2002). Verification strategies for establishing reliability and validity in qualitative research. *International Journal of Qualitative Methods*, *1*(2), 13.

Morse, J. M., & Niehaus, L. (2009). *Mixed method design: Principles and procedures* (Vol. 4): Left Coast Pr.

Müller, R., & Turner, J. R. (2007). Matching the project manager's leadership style to project type. *International Journal of Project Management, 25*(1), 21-32.

Musen, M. A. (1997). *Technology for Building Intelligent Systems: From Psychology to Engineering.* Paper presented at the Symposium on Motivation, Nebraska.

Muzaffar, Z., & Ahmed, M. A. (2010). Software development effort prediction: A study on the factors impacting the accuracy of fuzzy logic systems. *Information and Software Technology, 52*(1), 92-109.

Myers, M. D., & Tan, F. B. (2002). Beyond models of national culture in information systems research. *Journal of Global Information Management*, *10*(1), 24-32.

Nam, C. S., Lyons, J. B., Hwang, H. S., & Kim, S. (2009). The process of team communication in multi-cultural contexts: An empirical study using Bales' interaction process analysis (IPA). *International Journal of Industrial Ergonomics*, *39*(5), 771-782.

Nemecek, S., & Bemley, J. (1993). *A model for estimating the cost of AI software development: what to do if there are no lines of code?* Paper presented at the IEEE International Conference on Developing and Managing Intelligent System Projects, 1993.

New Zealand Career Services. (2010). *Who Earns What?* Wellington: New Zealand Government.

New Zealand Ministry of Business Innovation & Employment. (2013). *Information Communications and Technology*. Wellington: New Zealand Government.

State Sector Act, New Zealand. (1988).

Human Rights Act, New Zealand. (1993).

Nguyen, V., Steece, B., & Boehm, B. (2008). *A Constrained Regression Technique for COCOMO Calibration*.

Niazi, M., Babar, M., & Verner, J. (2010). Software Process Improvement barriers: A cross-cultural comparison. *Information and Software Technology, 52*(11), 1204-1216.

Nicoll, P. (2005). What lies ahead for public sector governance. *Keeping Good Companies, 57*, 19-26.

Niinimaki, T., Piri, A., Hynninen, P., & Lassenius, C. (2009). *Studying communication in agile software development: a research framework and pilot study*. Paper presented at the Proceedings of the ICMI-MLMI '09 Workshop on Multimodal Sensor-Based Systems and Mobile Phones for Social Computing.

Nilson, L. B. (2009). *The Graphic Syllabus and the Outcomes Map: Communicating Your Course*: Wiley.

Nixon, P., Harrington, M., & Parker, D. (2012). Leadership performance is significant to project success or failure: a critical analysis. *International Journal of Productivity & Performance Management*, *61*(2), 204-216.

Noether, G. E. (1981). Why Kendall Tau? *Teaching Statistics, 3*(2), 41-43.

Norušis, M. J. (2008). *SPSS 16.0 guide to data analysis*: Prentice Hall.

Norval, A. (2004). The politics of ethnicity and identity. In K. Nash & A. Scott (Eds.), *The Blackwell Companion to Political Sociology*. Malden, MA: Blackwell Publishing.

Ochieng, E. G., & Price, A. D. F. (2010). Managing cross-cultural communication in multicultural construction project teams: The case of Kenya and UK. *International Journal of Project Management, 28*(5), 449-460.

Ong, V., Richardson, D., Yanqing, D., Qile, H., & Johnson, B. (2009). The Role of Project Leadership in Achieving Effective Project Management. *Proceedings of the European Conference on Management, Leadership & Governance*, 157-163.

Onwuegbuzie, A. J., & Johnson, R. B. (2006). The Validity Issue in Mixed Research. *Research in the Schools, 13*(1), 48-63.

Onwuegbuzie, A. J., Slate, J. R., Leech, N. L., & Collins, K. M. T. (2007). Conducting mixed analyses: A general typology. *International Journal of Multiple Research Approaches, 1*(1), 4-17.

Ormerod, R. (1995). Putting Soft OR Methods to Work: Information Systems Strategy Development at Sainsbury's. *The Journal of the Operational Research Society, 46*(3), 277-293.

Orrego, A., Menzies, T., & El-Rawas, O. (2009). On the Relative Merits of Software Reuse. In Q. Wang, V. Garousi, R. Madachy & D. Pfahl (Eds.), *Trustworthy Software Development Processes, Proceedings* (Vol. 5543, pp. 186-197).

Orsman, B. (2010, 31 May 2010). Super City's set-up costs top $200m and counting. *New Zeland Herald*. Retrieved from http://www.nzherald.co.nz/politics/news/article.cfm?c_id=280&objectid=10648531

Osmundson, J. S., Michael, J. B., Machniak, M. J., & Grossman, M. A. (2003). Quality management metrics for software development. *Information & Management*, *40*(8), 799-812.

Oxford English Dictionary. (2013). *Oxford English Dictionary Online*. Retrieved 11 January 2013, from dictionary.oed.com

Page, S. E. (2007). *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*: Princeton University Press.

Panousopoulou, A., Galway, L., Nugent, C., & Parente, G. (2011). *SensorMed: A lightweight software library for pervasive healthcare systems.* Paper presented at the 24th International Symposium on Computer-Based Medical Systems (CBMS), 2011.

Papatheocharous, E., & Andreou, A. S. (2009). Classification and Prediction of Software Cost through Fuzzy Decision Trees. In J. Filipe & J. Cordeiro (Eds.), *Enterprise Information Systems* (Vol. 24, pp. 234-247). Berlin: Springer Berlin Heidelberg.

Parker, D., Craig, M., & Craig, M. A. (2008). *Managing Projects, Managing People*: Palgrave Macmillan.

Parkin, M. (1990). *Microeconomics*. Ontario: Addison Wesley.

Patton, M. (2002). *Qualitative research and evaluation methods*: Sage Publications, Inc.

Paul, S., Samarah, I. M., Seetharaman, P., & Mykytyn Jr, P. P. (2004). An Empirical Investigation of Collaborative Conflict Management Style in Group Support System-Based Global Virtual Teams. *Journal of Management Information Systems, 21*(3), 185-222.

Paulus, P., Nakui, T., Parthasarathy, N., & Baruah, J. (2004). *Preference for diverse workgroups and its relationship to perceptions and performance in diverse groups.* Paper presented at the 19th Society for Industrial and Organizational Psychology, Chicago.

Pavard, B., & Dugdale, J. (2006). The contribution of complexity theory to the study of socio-technical cooperative systems. In A. A. Minai & Y. Bar-Yam (Eds.), *Unifying Themes in Complex Systems* (pp. 39-48). Berlin: Springer.

Pearson, K. (1895). Contributions to the Mathematical Theory of Evolution. Note on Reproductive Selection. *Proceedings of the Royal Society of London, 59*(ArticleType: research-article / Full publication date: 1895 - 1896 / Copyright © 1895 The Royal Society), 300-305.

Pelled, L. H. (1996). Demographic Diversity, Conflict, and Work Group Outcomes: An Intervening Process Theory. *Organization Science, 7*(6), 615-631.

Pereira, J., Cerpa, N., Verner, J., Rivas, M., & Procaccino, J. D. (2008). What do software practitioners really think about project success: A cross-cultural comparison. *Journal of Systems and Software, 81*(6), 897-907.

Petter, S., & Gallivan, M. J. (2004). *Toward a Framework for Classifying and Guiding Mixed Method Research in Information Systems*. Paper presented at the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 8 - Volume 8.

Phelps, M. G. (2010). Comparing different estimates of productivity produced by the Office for National Statistics. *Economic & Labour Market Review, 4*(5), 25-29.

Piccoli, G., & Ives, B. (2003). Trust and the unintended effects of behavior control in virtual teams. *MIS Quarterly, 27*(3), 365-395.

Pieterse, V., Kourie, D. G., & Sonnekus, I. P. (2006). *Software engineering team diversity and performance*. Paper presented at the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering, 13*(3), 303-337.

Pinto, J. K. (2009). *Project Management: Achieving Competitive Advantage*: Prentice Hall.

Pinto, J. K., & Prescott, J. E. (1988). Variations in Critical Success Factors Over the Stages in the Project Life Cycle. *Journal of Management, 14*(1), 5-18.

Port, D., & McArthur, M. (1999). *A study of productivity and efficiency for object-oriented methods and languages.* Paper presented at the Software Engineering Conference, 1999. (APSEC '99) Proceedings. Sixth Asia Pacific.

Post, C., De Lia, E., DiTomaso, N., Tirpak, T. M., & Borwankar, R. (2009). Capitalizing on thought diversity for innovation. *Research Technology Management, 52*(6), 14-25.

Pressman, R. S. (2010). *Software engineering : a practitioner's approach* (3 ed.). New York: McGraw-Hill Higher Education.

Project Management Institute. (2008). *A Guide to the Project Management Body of Knowledge(PMBOK Guide)* (4th ed.): Project Management Institute.

Pruchno, R. A., Brill, J. E., Shands, Y., Gordon, J. R., Genderson, M. W., Rose, M., et al. (2008). Convenience Samples and Caregiving Research: How Generalizable Are the Findings? *Gerontologist, 48*(6), 820-827.

Putnam, L. H. (2001). *Software Life Cycle Model (SLIM)*. McLean: QSM.

Putnam, L. H., & Myers, W. (2003). *Five Core Metrics: Intelligence behind Successful Software Management*: Dorset House Publishing Co., Inc. New York, NY, USA.

Qureshi, H., & Rowlands, O. (2004). User satisfaction surveys and cognitive question testing in the public sector: the case of personal social services in England. *International Journal of Social Research Methodology, 7*(4), 273-287.

Rainey, H. G., Backoff, R. W., & Levine, C. H. (1976). Comparing Public and Private Organizations. *Public Administration Review, 36*(2), 233-244.

Ramo, S. (1989). National security and our technology edge. *Harvard Business Review, 67*(6), 115-120.

Ravishankar, K., & Navneet, B. (2006, Oct. 2006). *IT Application Assessment Model for Global Software Development.* Paper presented at the International Conference on Global Software Engineering, 2006.

Raykov, T., & Marcoulides, G. A. (2010). *Introduction to Psychometric Theory*: Taylor & Francis.

Reed, A. H., & Knight, L. V. (2010). Effect of a virtual project team environment on communication-related project risk. *International Journal of Project Management, 28*(5), 422-427.

Research and Markets. (December 2011). *Software: Global Industry Guide*. London: Author.

Riach, K. (2009). Exploring Participant-centred Reflexivity in the Research Interview. *Sociology, 43*(2), 356-370.

Richard, O. C., Barnett, T., Dwyer, S., & Chadwick, K. (2004). Cultural diversity in management, firm performance, and the moderating role of entrepreneurial orientation dimensions. *The Academy of Management Journal, 47*(2), 255-266.

Richard, O. C., Kochan, T. A., & McMillan-Capehart, A. (2002). The Impact of Visible Diversity on Organizational Effectiveness: Disclosing the Contents in Pandora's Black Box. *Journal of Business & Management, 8*(3), 265.

Richardson, P. (2005). Managing cultural diversity. *Engineering Management Journal, 15*(2), 24-27.

Richardson, R. M., & Smith, S. W. (2007). The influence of high/low-context culture and power distance on choice of communication media: Students' media choice to communicate with Professors in Japan and America. *International Journal of Intercultural Relations, 31*(4), 479-501.

Rocheleau, B., & Wu, L. (2002). Public Versus Private Information Systems. *The American Review of Public Administration, 32*(4), 379-397.

Rosh, L., Offermann, L. R., & Van Diest, R. (2011). Too close for comfort? Distinguishing between team intimacy and team cohesion. *Human Resource Management Review*.

Rubin, H. A. (1983). *Macro-estimation of software development parameters: The ESTIMACS system.* Paper presented at the Conference on software development tools, techniques and alternatives (July 25-28), Arlington, Vancouver.

Sala, F. (2002). Emotional competence inventory (eci): Technical manual. *Boston: Hay/McBer Group*.

Saliu, M. O., Ahmed, M., & AlGhamdi, J. (2004). *Towards adaptive soft computing based software effort prediction*.

Samovar, L. A., Porter, R. E., McDaniel, E. R., & Roy, C. S. (2012). *Communication Between Cultures*: Cengage Learning.

Sanders, J. M. (2002). Ethnic Boundaries and Identity in Plural Societies. *Annual Review of Sociology, 28*, 327-357.

Sands, R. G., Bourjolly, J., & Roer-Strier, D. (2007). Crossing Cultural Barriers in Research Interviewing. *Qualitative Social Work, 6*(3), 353-372.

Santos, F. S., & Moura, H. P. (2009). *What is wrong with the software development? Research trends and a new software engineering paradigm.* Paper presented at the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA, Orlando, FL.

Sawyer, S. (2001). A market-based perspective on information systems development. *Commun. ACM, 44*(11), 97-102.

Sawyerr, O. O., Strauss, J., & Yan, J. (2005). Individual value structure and diversity attitudes: The moderating effects of age, gender, race, and religiosity. *Journal of Managerial Psychology, 20*(6), 498-521.

Schrenker, R. A. (2006). Software engineering for future healthcare and clinical systems. *Computer, 39*(4), 26-32.

Schumpeter, J. (1911). *The Theory of Economic Development: An inquiry into profits, capital, credit, interest and the business cycle. 1934 translation* (R. Opie, Trans.). Cambridge, Mass: Harvard University Press.

Sebba, M., & Wootton, T. (1998). We, they and identity: Sequential versus identity-related explanation in code-switching. *Code-switching in conversation: Language, interaction and identity*, 262-286.

Sebt, M. H., Shahhosseini, V., & Rezaei, M. (2010). *Competency Based Optimized Assignment of Project Managers to Projects.* Paper presented at the 12th International Conference on Computer Modelling and Simulation (UKSim), 2010

Senior, P. A., & Bhopal, R. (1994). Ethnicity as a variable in epidemiological research. *BMJ: British Medical Journal, 309*(6950), 327.

Sethibe, T., Campbell, J., & McDonald, C. (2007). *IT Governance in Public and Private Sector Organisations: Examining the Differences and Defining Future Research Directions.* Paper presented at the 18th Australasian Conference on Information Systems, Toowoomba.

Shachaf, P. (2008). Cultural diversity and information and communication technology impacts on global virtual teams: An exploratory study. *Information and Management, 45*(2), 131-142.

Sharpe, R. (2001). "Globalization": The Next Tactic in the fifty-Year Struggle of Labor and Capital in Software Production. In R Baldoz, C Koeber & P. Kraft (Eds.), *The Critical Study of Work*. Philadelphia: Temple University Press.

Shee, H. K., & Pathak, R. D. (2006). Managing people and technology for enhancing competitiveness: Software industry perspective. *Journal of Transnational Management, 11*(1), 63-80.

Shepperd, M. (2007). *Software project economics: a roadmap*.

Sherif, M. (1966). *Group conflict and co-operation: their social psychology*: Routledge & K. Paul.

Sheskin, D. (2007). *Handbook of parametric and nonparametric statistical procedures*: Chapman & Hall/CRC.

Shibuya, B., & Tamai, T. (2009). *Understanding the Process of Participating in Open Source Communities.* Paper presented at the 2009 ICSE Workshop on Emerging Trends in Free-Libre-Open Source Software Research and Development, Vancouver.

Shukla, K. K. (2000). Neuro-genetic prediction of software development effort. *Information and Software Technology, 42*(10), 701-713.

Sibley, C. G., Liu, J. H., & Khan, S. S. (2008). Who are 'we'? Implicit associations between ethnic and national symbols for Maori and Pakeha in New Zealand. *New Zealand Journal of Psychology, 37*(2), 38-49.

Siegel, S., & Castellan, N. J. (1988). *Nonparametric statistics for the behavioral sciences*: McGraw-Hill.

Siemiatycki, M. (2008). *Managing optimism biases in the delivery of large-infrastructure projects: A corporate performance benchmarking approach.* Paper presented at the First International Conference on Infrastructure Systems and Services: Building Networks for a Brighter Future (INFRA), 2008.

Simons, H. (2009). *Case Study Research in Practice*: Sage.

Simpson, H. (2009). Productivity in public services. *Journal of Economic Surveys, 23*(2), 250-276.

Simpson, L., & Akinwale, B. (2007). Quantifying stability and change in ethnic group. *JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-, 23*(2), 185.

Sinha, A., Lahiri, R. N., Chowdhury, S., Chowdhury, S. P., & Song, Y. H. (2007, 4-6 Sept. 2007). *Complete IT solution for Enterprise Asset Management (EAM) in Indian power utility business.* Paper presented at the Universities Power Engineering Conference, 2007. UPEC 2007. 42nd International.

Skevoulis, S., Campedelli, J., Holdsworth, K., Verel, J., & Tavales, S. (2009). *Engineering a Software Supported Health Risk Appraisal Method: A Joint Effort between Academia and Health Care Industry.* Paper presented at the 22nd Conference on Software Engineering Education and Training, 2009. CSEET '09. .

Skulmoski, G. J., & Hartman, F. T. (2010). Information systems project manager soft competencies: A project-phase investigation. *Project Management Journal*, *41*(1), 61-80.

Skyttner, L. (2005). *General systems theory: problems, perspectives, practice*. Singapore: World Scientific Pub Co Inc.

Sligo, F., Fountaine, S., O'Neill, D., & Sayers, J. (2000). *Effective Communication in Business Communication.* Palmerston North, NZ: Software Technology New Zealand Ltd.

Sligo, F., & McLean, K. (2000). *Effective Communication in Business*: Software Technology NZ, Limited.

Smart, A., Tutton, R., Martin, P., Ellison, G. T. H., & Ashcroft, R. (2008). The Standardization of Race and Ethnicity in Biomedical Science Editorials and UK Biobanks. *Social Studies of Science, 38*(3), 407-423.

Šmite, D., Wohlin, C., Gorschek, T., & Feldt, R. (2010). Empirical evidence in global software engineering: a systematic review. *Empirical Software Engineering*, *15*(1), 91-118.

Smith, A. (1986). *The ethnic origins of nations*. Hoboken: Wiley-Blackwell.

Smith, B. L. (2002). Software development cost estimation for infrastructure systems. *Journal of Management in Engineering*, *18*(3), 104-110.

Smith, H. W. (1975). *Strategies of social research: the methodological imagination*: Prentice-Hall.

Snow, A. P., & Keil, M. (2002). The challenge of accurate software project status reporting: a two-stage model incorporating status errors and reporting bias. *Engineering Management, IEEE Transactions on*, *49*(4), 491-504.

Snowdeal-Carden, B. A. (2013). *Investigating team cohesion in COCOMO II.2000.* Unpublished Ph.D., Capella University, Ann Arbor.

Soffer, P., & Hadar, I. (2007). Applying ontology-based rules to conceptual modeling: a reflection on modeling decision making. *Eur J Inf Syst, 16*(5), 599-611.

Sommerville, I. (2004). Software Process Models. In A. Tucker (Ed.), *Computer Science Handbook.* . Florida: Chapman & Hall/CRC

Sprent, P., & Smeeton, N. C. (2007). *Applied nonparametric statistical methods*: Chapman & Hall/CRC.

Standish Group. (1995). *The Chaos Report*. West Yarmouth: The Standish Group International.

Standish Group. (2001). *Extreme Chaos*. West Yarmouth: The Standish Group International.

State Services Commission of New Zealand. (2008). Equality and Diversity: New Zealand Public Service Equal Employment Opportunities Policy. Retrieved 26 July 2012, 2012, from http://www.ssc.govt.nz/public-service-eeopolicy

Statistics New Zealand. (2005a). Ethnicity New Zealand Standard Classification 2005, Version 1.0, Effective Date: 01/06/2005. Retrieved 11 September 2008, from http://www.stats.govt.nz/surveys_and_methods/methods/classifications-and-standards/classification-related-stats-standards/ethnicity.aspx

Statistics New Zealand. (2005b). Statistical Standard for Ethnicity 2005. Retrieved 18 May 2010, from http://www.stats.govt.nz/~/media/Statistics/browse-categories/population/census-counts/review-measurement-ethnicity/statistical-standard-for-ethnicity-2005.pdf

Statistics New Zealand. (2009). *Information and Communication Technology in New Zealand and Australia.* . Wellington: Statistics New Zealand.

Statistics New Zealand. (2012). *IT is it: economy's appetite for computer skills grows*. Wellington: Statistics New Zealand.

Statistics New Zealand. (2013). *2013 Census QuickStats about national highlights*. Wellington: Statistics New Zealand.

Stewart, G. (2006). A meta-analytic review of relationships between team design features and team performance. *Journal of Management, 32*(1), 29.

Stigler, S. (2008). Fisher and the 5% level. *Chance, 21*(4), 12-12.

Stober, T., & Hansmann, U. (2009). *Agile Software Development*. Berlin: Springer.

Strode, D. E., Hope, B., Huff, S., & Link, S. (2011). *Coordination Effectiveness In An Agile Software Development Context.* Paper presented at the PACIS 2011, Brisbane.

Sujin, K. H. (2005). The Compositional Impact of Team Diversity on Performance: Theoretical Considerations. *Human Resource Development Review, 4*(2), 219-245.

Sukhoo, A., Barnard, A., Eloff, M. M., & Van Der Poll, J. A. (2005). Accommodating Soft Skills in Software Project Management. *Informing Science: International Journal of an Emerging Transdiscipline, 2*, 691-703.

Tadmor, C. T., Satterstrom, P., Jang, S., & Polzer, J. T. (2012). Beyond Individual Creativity: The Superadditive Benefits of Multicultural Experience for Collective Creativity in Culturally Diverse Teams. *Journal of Cross-Cultural Psychology*.

Taeho, L., Donoh, C., & Jongmoon, B. (2010). *Empirical study on enhancing the accuracy of software cost estimation model for defense software development project applications.* Paper presented at the The 12th International Conference on Advanced Communication Technology (ICACT), 2010.

Tan, T., Li, Q., Boehm, B., Yang, Y., He, M., & Moazeni, R. (2009). *Productivity trends in incremental and iterative software development.* Paper presented at the 2009 3rd International

Symposium on Empirical Software Engineering and Measurement, ESEM 2009, Lake Buena Vista, FL.

Taras, V., Kirkman, B. L., & Steel, P. (2010). Examining the impact of< em> Culture's consequences: A three-decade, multilevel, meta-analytic review of Hofstede's cultural value dimensions. *Journal of Applied Psychology, 95*(3), 405.

Tashakkori, A., & Teddlie, C. (1998). *Mixed methodology: Combining qualitative and quantitative approaches*. Thousand Oaks: Sage Publications, Inc.

Tashakkori, A., & Teddlie, C. (2003). *Handbook of mixed methods in social & behavioral research*: Sage Publications, Inc.

Tashakkori, A., & Teddlie, C. (2008). *Foundations of Mixed Methods Research: Integrating Quantitative and Qualitative Approaches in the Social and Behavioral Sciences*: SAGE Publications.

Tashakkori, A., & Teddlie, C. (2010). *Sage handbook of mixed methods in social & behavioral research*: Sage Publications, Inc.

Taylor, H., & Woelfer, J. P. (2009). *Critical skills for IT project management and how they are learned*. Paper presented at the Proceedings of the special interest group on management information system's 47th annual conference on Computer personnel research.

Teddlie, C., & Yu, F. (2007). Mixed Methods Sampling. *Journal of Mixed Methods Research, 1*(1), 77-100.

The Economist. (2011). Pipe dreams (How to track down water leaks). *The Economist* (4 June 2011).

Thite, M. (1999). Identifying key characteristics of technical project leadership. *Leadership & Organization Development Journal, 20*(5), 253-261.

Thomas, J., & Mullaly, M. (2007). Understanding the value of project management: First steps on an international investigation in search of value. *Project Management Journal, 38*(3), 74-89.

Thomsett, R. (2002). *Radical Project Management*: Prentice Hall PTR.

Tiatia, J. (2008). *Pacific cultural competencies: A literature review*: Ministry of Health Wellington, New Zealand.

Tidd, J., & Bessant, J. (2011). *Managing Innovation: Integrating Technological, Market and Organizational Change*: Wiley.

Tiwana, A. (2008). Impact of classes of development coordination tools on software development performance: A multinational empirical study. *ACM Trans. Softw. Eng. Methodol., 17*(2), 1-47.

Todd, P., & Benbasat, I. (1987). Process Tracing Methods in Decision Support Systems Research: Exploring the Black Box. *MIS Quarterly, 11*(4), 493-512.

Trade Me. (2012). *Salary Guide*. Wellington.

Trauth, E., & Jessup, L. M. (2000). Understanding computer-mediated discussions: positivist and interpretive analyses of group support system use. *MIS Quarterly, 24*(1), 43-79.

Trauth, E., Quesenberry, J., & Huang, H. (2009). Retaining women in the US IT workforce: theorizing the influence of organizational factors. *European Journal of Information Systems, 18*(5), 476-497.

Trauth, E., Quesenberry, J. L., & Huang, H. (2006). *Cross-cultural influences on women in the IT workforce*. Paper presented at the Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research: Forty four years of computer personnel research: achievements, challenges and the future.

Triandis, H. C. (2004). The Many Dimensions of Culture. *The Academy of Management Executive (1993-2005), 18*(1), 88-93.

Trompenaars, F., & Hampden-Turner, C. (2004). *Managing People Across Cultures*: Wiley.

Turner, J. R. (2008). *The Handbook of Project-based Management: Leading Strategic Change in Organizations*: McGraw-Hill Companies,Incorporated.

Turner, J. R., & Müller, R. (2005). The project manager's leadership style as a success factor on projects: a literature review. *Project Management Journal, 36*(2), 49-61.

Turner, L. (2009). Gender diversity and innovative performance. *International Journal of Innovation and Sustainable Development, 4*(2-3), 123-134.

U.S. Department of Labor. (2012). *Occupational Outlook Handbook, 2012-2013 Edition*. Retrieved from http://www.bls.gov/ooh/.

Unger, B., & Walker, S. (1977). *Improving team productivity in system software development.* Paper presented at the Fifteenth annual SIGCPR conference 1977.

United States Equal Employment Opportunity Commission. (18 November 2004). *EEOC Agrees to Landmark Resolution of Discrimination Case Against Abercrombie & Fitch*. LOS ANGELES: United States Equal Employment Opportunitu Commission,.

Van Dick, R., Van Knippenberg, D., Hagele, S., Guillaume, Y. R. F., & Brodbeck, F. C. (2008). Group diversity and group identification: The moderating role of diversity beliefs. *Human Relations, 61*(10), 1463-1492.

van Knippenberg, D. (2007). *Understanding Diversity*. Rotterdam: Erasmus Research Institute of Management.

van Knippenberg, D., Dawson, J. F., West, M. A., & Homan, A. C. (2011). Diversity faultlines, shared objectives, and top management team performance. *Human Relations*, *64*(3), 307-336.

van Knippenberg, D., De Dreu, C. K. W., & Homan, A. C. (2004). Work group diversity and group performance: An integrative model and research agenda. *Journal of Applied Psychology, 89*(6), 1008.

van Knippenberg, D., & Schippers, M. C. (2007). Work group diversity. *Annu. Rev. Psychol., 58*, 515-541.

Venkatesh, V., Brown, S., & Bala, H. (2013). Bridging the qualitative–quantitative divide: Guidelines for conducting mixed methods Research in information systems. *MIS Quarterly, 37*(1), 21-54.

Verner, J. M., Sampson, J., Tosic, V., Bakar, N. A. A., & Kitchenham, B. A. (2009). *Guidelines for industrially-based multiple case studies in software engineering.* Paper presented at the Third International Conference on Research Challenges in Information Science, 2009. RCIS 2009. .

Vilvovsky, S. (2008). *Differences between public and private IT outsourcing: common themes in the literature*. Paper presented at the Proceedings of the 2008 international conference on Digital government research.

Wageman, R., Fisher, C. M., & Hackman, J. R. (2009). Leading Teams When the Time is Right: Finding the Best Moments to Act. *Organizational Dynamics, 38*(3), 192-203.

Wagner Iii, J. A. (1995). Studies of individualism-collectivism: effects on cooperation in groups. *Academy of Management Journal, 38*(1), 152-173.

Walsham, G. (1995). The Emergence of Interpretivism in IS Research. *Information Systems Research, 6*(4), 376-394.

Walsham, G. (2002). Cross-cultural software production and use: A structurational analysis. *MIS Quarterly: Management Information Systems, 26*(4), 359-380.

Wang, Y. (2009). *Building the linkage between project managers' personality and success of software projects*. Paper presented at the Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement.

Ward, C., & Liu, J. (2012). Multiculturalism in Europe and New Zealand. On *Sunday Morning*. Wellington: Radio New Zealand.

Watson, W. E., Kumar, K., & Michaelsen, L. K. (1993). Cultural diversity's impact on interaction process and performance: Comparing homogeneous and diverse task groups. *The Academy of Management Journal*, *36*(3), 590-602.

Weinberg, G. (1998). *The psychology of computer programming*: Dorset House.

Wellington, C. A., Briggs, T., & Girard, C. D. (2005). *Examining team cohesion as an effect of software engineering methodology*. Paper presented at the ACM SIGSOFT Software Engineering Notes.

Whitworth, E., & Biddle, R. (2007). *Motivation and cohesion in agile teams*. Paper presented at the Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming.

William, J. G., & Andrew, B. (2009). *6 Generating data through questions and observations. Working with Qualitative Data. SAGE Publications, Ltd*. London, UK: Sage Publications.

Williams, J. E. (2009). *Connecting people : investigating a relationship between internet access and social cohesion in local community settings.* Massey University, Palmerston North.

Williams, K. Y., & O'Reilly, C. A., III. (1998). Demography and diversity in organizations: a review of 40 years of research. *Research in Organizational Behavior, 20*, 77.

Wilson, J. (2010). *Essentials of Business Research: A Guide to Doing Your Research Project*: Sage Publications.

Winkler, V. A., & Bouncken, R. B. (2009). *Cultural diversity in global innovation teams: Linking effects of cultural diversity to the innovation process.* Paper presented at the PICMET: Portland International Center for Management of Engineering and Technology.

Wong, B., & Bhatti, M. (2009). *The influence of team relationships on software quality.* Paper presented at the ICSE Workshop on Software Quality, 2009.

Woolley, A. W., Gerbasi, M. E., Chabris, C. F., Kosslyn, S. M., & Hackman, J. R. (2008). Bringing in the Experts: How Team Composition and Collaborative Planning Jointly Shape Analytic Effectiveness. *Small Group Research, 39*(3), 352-371.

Xia, W., Ho, D., & Capretz, L. (2008). A neuro-fuzzy model for function point calibration. *WSEAS Transactions on Information Science and Applications, 5*(1), 22-30.

Yang, L.-R., Huang, C.-F., & Wu, K.-S. (2011). The association among project manager's leadership style, teamwork and project success. *International Journal of Project Management, 29*(3), 258-267.

Ye, Y., Yamamoto, Y., & Nakakoji, K. (2007). *A socio-technical framework for supporting programmers.* Paper presented at the The 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering Dubrovnik, Croatia.

Yin, R. K. (2003). *Case study research: Design and methods* (3rd ed.). Thousand Oaks: Sage.

Yin, R. K. (2009). *Case study research: Design and methods* (4th ed.). Thousand Oaks: Sage.

Yu, H., Bao, X., & Yang, S. (2009). *Research and Improvement of Team Software Process.* Paper presented at the WRI World Congress on Computer Science and Information Engineering, 2009.

Zijiang, Y., & Paradi, J. C. (2009). *A DEA evaluation of software project efficiency*. Paper presented at the IEEE International Conference on Industrial Engineering and Engineering Management, 2009. IEEM 2009.

Zowghi, D., & Bargi, A. (2011, Aug. 29 2011-Sept. 2 2011). *Software versus IT Service: A Comparative Study from Requirements Engineering Perspective*. Paper presented at the 22nd International Workshop on Database and Expert Systems Applications (DEXA), 2011.

Zsembik, B. A. (1994). Ethnic and sociodemographic correlates of the use of proxy respondents: The National Survey of. *Research on Aging*, *16*(4), 401.

APPENDICES

Appendix A – Covering Email for Invitation to Participate

This appendix shows the covering email for the initial invitation used for this study.

```
Dear [Invitee’s first name],

I am a PhD student with the School of Management at Massey University in Wellington.
For my PhD, I am looking at the role of team diversity and other factors in the
productivity of software development in New Zealand. To do this, I am analysing
software projects completed in New Zealand. Prior to beginning my PhD a year ago, I
spent 20 years working in software development, including 10 years in management
roles.

I would like to invite [invitee’s organisation] to participate in this research
which looks at recently completed software development projects. Only a small amount
of time is required (that is, a one hour interview with the project manager) and
access to some system and project documentation. The results would be confidential
with only the aggregated results published in the thesis and any other publications.
In return, I could provide [invitee’s organisation] with a confidential report on
the productivity achieved in the software projects at [invitee’s organisation], the
relevant industry benchmarks for software productivity and an analysis of the
primary factors that appeared to influence productivity based on established
industry models.

The attached one page “invitation to participate” provides more detail.

I look forward to hearing from your response to this invitation. I would be happy to
call you or meet with you to discuss this further if that would be useful.

Kind Regards,
Jules Congalton

***************************************
Jules Congalton
PhD Candidate
School of Management
Massey University
Private Bag 756
Wellington

T: (04) 027 224 3686
F: (04) 802 0290
E: j.congalton@massey.ac.nz
***************************************
```

Figure 8. Invitation email

Appendix B – Invitation to Participate

This appendix shows the invitation to participate used for this study.

Massey University
WELLINGTON

Private Box 756
Wellington 6140
New Zealand
T 64 4 801 5799
F 64 4 801 2692
www.massey.ac.nz

[Date]

[Invitee's full name]
[Invitee's address]

Dear [Invitee's first name],

Subject: Invitation to Participate in Research on Factors Affecting Software Development Productivity

This letter is to invite your organisation to participate in research regarding the factors affecting software development productivity in New Zealand. The specific goal of this research is to understand how diversity within software development teams affects the productivity those teams achieve. In order to isolate the effects of diversity, data will also be gathered on all factors affecting software development productivity.

I will document the results of this research as the thesis for my PhD in Information Systems which I am currently undertaking at Massey University in Wellington. In that thesis, and in any other publication arising from this research, care will be taken to ensure participating organisations cannot be identified. All research data will be held securely, encrypted and password protected. Only myself and the two supervisors for this research will have access to your organisation's data. The supervisors overseeing this research are Dr Keri Logan and Dr Barbara Crump from the School of Management at Massey University in Wellington. Further details of security and ethical considerations will be provided in a confidentiality agreement and information sheet.

If you agree to participate, the involvement required from your organisation would be access to information your organisation holds on the software development projects you nominate and a one hour interview with each project manager for the nominated projects. In return for your participation with this study, I can provide you with a confidential report for your organisation providing an analysis of the software projects at your organisation. This report would show the level of productivity achieved on those projects, how that compares to other similar projects internationally and an analysis of what factors contributed to, or hindered productivity in the software projects analysed.

As your organisation is involved in the development of software, your organisation has valuable information that will contribute to the overall findings of this research. Therefore, we would greatly appreciate your participation in this valuable research.

Thank for your time.

Yours sincerely,

J. C

Jules Congalton
PhD Candidate
School of Management

Figure 9. Invitation to participate

Appendix C – Information Sheet

This appendix shows the information sheet used for this study.

Massey University
WELLINGTON

Private Box 756
Wellington 6140
New Zealand
T 64 4 801 5799
F 64 4 801 2692
www.massey.ac.nz

Information Sheet

INFORMATION FOR RESEARCH PARTICIPANTS

Title of Research Project: The Role of Team Diversity in Software Development Productivity in New Zealand.

Name of Researcher: Jules Congalton, Doctoral Student
School of Management, Massey University, Wellington, New Zealand

Researcher Introduction

This research is being carried out by Jules Congalton, who is currently undertaking a PhD in Information Systems at Massey University with an expected completion date in 2012. Jules Congalton has previously worked in the software development industry for over 20 years including a number of roles in both the public and private sectors.

Project Description

Team composition has been found to be a major influence of team performance which raises the question of how ethnic and gender diversity within software development teams affects the performance of those teams. Therefore, this research seeks to understand how diversity in software development teams influences productivity that those teams achieve. Data is being gathered about software projects completed in the last 12 months from software-producing organisations within New Zealand. Information will be collected using interviews with project managers augmented by documentation already held by the organisation. A detailed analysis will be undertaken to reveal themes and patterns regarding the relationships between team diversity and software development productivity.

Ethics

This research is being undertaken in accordance with the "Massey University Code of Ethical Conduct for Research, Teaching and Evaluations Involving Human Participants" (Massey University, 2010). This project has been evaluated by peer review and judged to be low risk. Consequently, it has not been reviewed by one of the University's Human Ethics Committees. The researcher named above is responsible for the ethical conduct of this research.

If you have any concerns about the conduct of this research that you wish to raise with someone other than the researcher or supervisor, please contact the Research Ethics Office, telephone 06 350 5573, email humanethics@massey.ac.nz.

Participants' Rights

Participants are under no obligation to accept an invitation to participate. If they do decide to participate, they have the right to:

- decline to answer any particular question;
- withdraw from the study;
- ask any questions about the study at any time during participation;
- provide information on the understanding that their name will not be used unless they give permission to the researcher;
- be given access to a summary of the project findings when it is concluded.

If recording occurs during interviews, participants may ask for the recorder to be turned off at any time during the interview.

Project Contacts

The researcher undertaking this project, Jules Congalton, can be contacted through the School of Management, Massey University, Private Bag 756, 63 Wallace Street, Wellington 6140, phone 027 224 3686, or via email: j.congalton@massey.ac.nz.

The primary supervisor for this project is Dr Keri Logan of the Department of Management, Massey University, who can be contacted at Private Bag 756, 63 Wallace Street, Wellington 6140, phone (04) 801 5799 ext 6366, or via email: k.a.logan@massey.ac.nz. Participants are invited to contact the researcher and/or supervisor if they have any questions about the project.

Figure 10. Information sheet

Appendix D – Consent Form

This appendix shows the consent form used for this study.

Massey University
WELLINGTON

Private Box 756
Wellington 6140
New Zealand
T 64 4 801 5799
F 64 4 801 2692
www.massey.ac.nz

Consent Form

CONSENT TO PARTICIPATE IN A RESEARCH PROJECT

Title of Research Project: The Role of Team Diversity in Software Development Productivity in New Zealand.

Name of Researcher: Jules Congalton
Doctoral Student
School of Management
Massey University
Wellington, New Zealand

PARTICIPANT:
I have read the attached Information Sheet and understand it. I understand I have the opportunity to ask questions and that at my request I will receive feedback on the outcome of the research at its conclusion. I understand the data will be retained for five years (as required by university regulations) and will be held securely.

I understand that the published results of this research project will not identify me personally or my organisation.

I understand that I may withdraw myself or any information I have provided from this project before data collection is completed, without having to given reasons and without penalty of any sort.

I consent to take part in this research project having received details of the research, what my participation will involve and an explanation of how the published results will be used.

Signature: ______________________ **Date:** _______

Name of Participant: ______________________

RESEARCHER:
The participant has received details of the research, what their participation will involve and an explanation of how the published results will be used. I have provided a contact email address and phone number should the participant require further information

Signature: ______________________ **Date:** ....

Figure 11. Consent form

Appendix E – Request for Documentation

This appendix shows the request for documentation used for this study.

Massey University
WELLINGTON

Private Box 756
Wellington 6140
New Zealand
T 64 4 801 5799
F 64 4 801 2692
www.massey.ac.nz

Request for Project and System Documentation

Title of Research Project: The Role of Team Diversity in Software Development Productivity in New Zealand.

Name of Researcher: Jules Congalton, Doctoral Student
School of Management, Massey University, Wellington, New Zealand

Introduction

These documents are being requested to support the above research project. The information sheet for this research project contains further information about the research and how this information will be used. The documents listed below are requested to be sent to the research either electronically or in printed form. Contact details are below.

Documents Requested

The following documents are requested:

Document type	Type of information required	Documentation requested that may contain the required information
Project documentation	Information about the project such as total project effort, project duration and the challenges that were encountered during the project	1. Project overview / project brief / project scope / change request 2. Project status reports 3. Post Implementation Review 4. Timesheet summary, (ideally total hours of project effort by activity, e.g. requirements, design, testing, etc) 5. Issues and risk registers 6. Project methodology / process definition
System documentation	Information about the system that was developed or enhanced such as its purpose, size and complexity	7. Requirements and design specifications covering the changes made by the project – especially the design of the inputs, outputs and interfaces 8. Data model - ideally showing what was changed by the project 9. User guides 10. Function point count for the changes made by project 11. Total lines of source code added, changed and deleted by the project

Contact Details

The researcher undertaking this project, Jules Congalton, can be contacted through the School of Management, Massey University, Private Bag 756, 63 Wallace Street, Wellington 6140, phone 027 224 3686, or via email: j.congalton@massey.ac.nz.

Figure 12. Request for documentation

Appendix F – List of Variables

In order to understand how ethnic diversity influenced productivity, it was first necessary to identify what factors overall were most significant in influencing productivity. A total of 42 variables could potentially be captured for each project and each variable has been assigned a three or four letter character code. These variables operationalise the various components of the conceptual model and can be divided into two sets. The first set is the COCOMO II productivity factors (Boehm, et al., 2000b) and the second set is the additional variables that were considered to be potentially relevant to the analysis of diversity and productivity. The first set of variables is described in Table 22.

Table 22. Definition of COCOMO II variables (Boehm, et al., 2000)

Variable Name	Code	Description
Analyst Capability	ACAP	Analysts are personnel who work on requirements, high-level design and detailed design. The major attributes considered in this rating are analysis and design ability, efficiency and thoroughness, and the ability to communicate and co-operate. The rating does not consider the level of experience of the analyst; that is rated with APEX, LTEX, and PLEX.
Applications Experience	APEX	The rating for this factor is dependent on the level of applications experience of the project team developing the software system or subsystem. The ratings are defined in terms of the project team’s equivalent level of experience with this type of application.
Product Complexity	CPLX	Complexity is the combination of the complexity of the following five areas: control operations, computational operations, device-dependent operations, data management operations, and user interface management operations.
Data Base Size	DATA	This factor attempts to capture the effect large test data requirements have on product development. The rating is determined by calculating the ratio of bytes in the testing database to SLOC in the program. The reason the size of the database is important to consider is because of the effort required to generate the test data that will be used to exercise the program.
Documentation Match to Life-Cycle Needs	DOCU	The suitability of the project’s documentation to its life-cycle needs. The rating scale goes from Very Low (many life-cycle needs uncovered
Development Flexibility	FLEX	The degree of need for software conformance with pre-established requirements or external interface requirements.
Hours of effort	HRS	The total number of hours of effort from all team members to complete the project.

Variable Name	Code	Description
Language used	LANG	The development language used
Language and Tool Experience	LTEX	This is a measure of the level of programming language and software tool experience of the project team developing the software system or subsystem. Software development includes the use of tools that perform requirements and design representation and analysis, configuration management, document extraction, library management, program style and formatting, consistency checking, planning and control. In addition to experience in the project's programming language, experience on the project's supporting tool set also affects development effort. A low rating is given for experience of less than 2 months. A very high rating is given for experience of 6 or more years.
Programmer Capability	PCAP	An evaluation of the capability of the programmers as a team rather than as individuals. Major factors considered are ability, efficiency and thoroughness, and the ability to communicate and co-operate. The experience of the programmer is not considered here; it is rated with APEX, LTEX, and PLEX.
Personnel Continuity	PCON	The project's annual personnel turnover: ranging from less than 3% per year (very high continuity
Platform Experience	PLEX	This represents the teams understanding of the use of the platforms being used such as graphic user interface, database, networking, and distributed middleware capabilities.
Process Maturity	PMAT	This factor is the maturity of the processes being used for the project. The process maturity is defined according the Software Engineering Institute's Capability Maturity Model (CMM
Precedentedness	PREC	If a project being undertaken is similar to previous projects, then the precedentedness is high.
Platform Volatility	PVOL	"Platform" is used here to mean the complex of hardware and software (such as OS and DBMS).
Required Software Reliability	RELY	This is the measure of the extent to which the software must perform its intended function over a period of time. If the effect of a software failure is only slight inconvenience then RELY is very low. If a failure would risk human life then RELY is very high.
Architecture / Risk Resolution	RESL	This covers the degree of design thoroughness and risk elimination through product and project reviews.
Requirements volatility	REVL	The degree to which the requirements changed throughout the projects. This is effectively a measure of the quality of the requirements.
Developed for Reusability	RUSE	This factor accounts for the additional effort needed to construct components intended for reuse on current or future projects.
Required Development Schedule	SCED	This rating measures the schedule constraint imposed on the project team developing the software. The ratings are defined in terms of the percentage of schedule stretch-out or acceleration with respect to a nominal schedule for a project requiring a given amount of effort. Accelerated schedules tend to produce more

Variable Name	Code	Description
		effort in the earlier phases to eliminate risks and refine the architecture, more effort in the later phases to accomplish more testing and documentation in parallel.
Multisite Development	SITE	This factor is rated using the assessment and judgement-based averaging of two factors: site collocation (from fully collocated to international distribution
Product Size	SIZE	This factor captures the size of the software being developed. The sizing method used for the research defined here is function points – more specifically unadjusted function points (UFP
Source Lines of Code per Function Point	SPF	The average number of lines of source code to deliver one UFP depends on the software implementation language used. For example, according to (Jones, 1996
Main Storage Constraint	STOR	This rating represents the degree of main storage constraint imposed on a software system.
Team Cohesion	TEAM	This factor accounts for the sources of project turbulence and entropy because of difficulties in synchronising the project's stakeholders: users, customers, developers, maintainers, interfacers, others. These difficulties may arise from differences in stakeholder objectives and cultures; difficulties in reconciling objectives; and stakeholders' lack of experience and familiarity in operating as a team.
Execution Time Constraint	TIME	This is a measure of the execution time constraint imposed upon a software system. The factor is rated in terms of the percentage of available execution time expected to be used by the system consuming the execution time resource.
Use of Software Tools	TOOL	The capability, maturity, and integration of the use of tools. The tool rating ranges from simple edit and code (Very Low) to an integrated and mature tool set (Very High).
Unadjusted Function Points	UFP	The size of the software change, or of the software created. This is measure in unadjusted function points.

The additional variables that have been used in this study and which are not defined in COCOMO II are shown in Table 23.

Table 23. Definition of variables not in COCOMO II

Name	Code	Description
Use of agile practices	AGIL	This is an assessment of the degree to which agile practices were used. This ranges from none, through to extensive use of agile practices.
Number of ethnicities in the team	ETH	A count of the number of ethnicities in the team. Ethnicities are either self-identified or estimated by the project manager. The top level of the NZ statistics standard list of ethnicities is used, which

Name	Code	Description
		gives five possible ethnicities. *Appendix G – List of Ethnic Groups* provides a full list of all ethnicities.
Software producer?	SWPR	Whether the organisation's primary activity is software production. A 1 indicates it is, a 0 indicates it is not.
Normalised Project Delivery Rate	NPDR	This is a measure of software development productivity and is a ratio of input to outputs where inputs are measured in hours of effort and outputs are measure in function points (P. Hill, 2010). The PDR is therefore the hours it takes to deliver each function point where a lower figure indicates a more productive project. This allows a comparison of productivity across the sample of projects analysed and with industry benchmarks (International Software Benchmarking Standards Group, 2009). This has been normalised to improved comparability between projects as not all projects reported hours for all project phases.
Ethnic diversity index	BEDI	A measure of ethnic diversity in the team based on Blau's index of heterogeneity (Blau, 1977). This is the probability of two people selected at random being from different ethnicities. The range of values could in theory be from 0 (all one ethnicity) through to 1 (a team of five people with one of each of the five ethnic groups). Richard et al. (2002) suggests that a value below 0.25 is low diversity, 0.25 to .50 implies moderate diversity and a value above 0.5 represents a high level of diversity.
PM's gender	PMGN	The gender of the project manager.
PM's age	PMAG	The age group of the project manager, for example, 30 – 39, 40-49, etc.
PM years	PMYR	The number of years the project manager has managing projects
Project length	PLNG	The length of the project in months.
Government?	GOVT	Whether the project was undertaken by a government agency. A 1 indicates it is, a 0 indicates it is not.
Project success as rated by the project manager	PRSC	The success of the project as rated by the project manager. This can be Major problems, Successful or Very successful.
Project productivity as rated by the project manager	PRPR	The productivity of the project as rated by the project manager. This can be from very low through to very high, with very low indicating poor productivity and very high indicating a highly productive project.
Enhancement or New Development	TYPE	Whether the project was undertaken by a government agency. A 1 indicates it is, a 0 indicates it is not.
Team size	TMSZ	The number of people in the team. The project manager was included in TMSZ when they were an active part of the team. As a guide, if they spent more than 20% of their time on the project, then they were a team member

Appendix G – List of Ethnic Groups

This appendix shows the list of ethnic groups used for this study which is reproduced from *Ethnicity New Zealand Standard Classification* (Statistics New Zealand, 2005a).

Table 24. Ethnic groups

Level 1	**Level 2**	**Level 3**	**Level 4**	
1	**European**			
	11	New Zealand European		
		111	New Zealand European	
			11111	New Zealand European
	12	Other European		
		121	British and Irish	
			12100	British nfd
			12111	Celtic nfd
			12112	Channel Islander
			12113	Cornish
			12114	English
			12115	Gaelic
			12116	Irish
			12117	Manx
			12118	Orkney Islander
			12119	Scottish
			12120	Shetland Islander
			12121	Welsh
			12199	British nec
		122	Dutch	
			12211	Dutch
		123	Greek	
			12311	Greek
		124	Polish	
			12411	Polish
		125	South Slav	
			12500	South Slav nfd
			12511	Croatian
			12512	Dalmatian
			12513	Macedonian
			12514	Serbian
			12515	Slovenian
			12516	Bosnian
			12599	South Slav nec
		126	Italian	
			12611	Italian

		127	German	
			12711	German
		128	Australian	
			12811	Australian
		129	Other European	
			12911	Albanian
			12912	Armenian
			12913	Austrian
			12914	Belgian
			12915	Bulgarian
			12916	Belorussian
			12917	Corsican
			12918	Cypriot nfd
			12919	Czech
			12920	Danish
			12921	Estonian
			12922	Finnish
			12923	Flemish
			12924	French
			12925	Greenlander
			12926	Hungarian
			12927	Icelandic
			12928	Latvian
			12929	Lithuanian
			12930	Maltese
			12931	Norwegian
			12932	Portuguese
			12933	Romanian
			12934	Gypsy
			12935	Russian
			12936	Sardinian
			12937	Slavic
			12938	Slovak
			12939	Spanish
			12940	Swedish
			12941	Swiss
			12942	Ukrainian
			12943	American
			12944	Burgher
			12945	Canadian
			12946	Falkland Islander
			12947	New Caledonian
			12948	South African nec
			12949	Afrikaner
			12950	Zimbabwean
			12999	European nec
2	**Māori**			
	21	Māori		
		211	Māori	

			21111	Māori
3	**Pacific Peoples**			
	30	Pacific Peoples nfd		
		300	Pacific Peoples nfd	
			30000	Pacific Peoples nfd
	31	Samoan		
		311	Samoan	
			31111	Samoan
	32	Cook Islands Maori		
		321	Cook Islands Maori	
			32100	Cook Islands Maori nfd
			32111	Aitutaki Islander
			32112	Atiu Islander
			32113	Mangaia Islander
			32114	Manihiki Islander
			32115	Mauke Islander
			32116	Mitiaro Islander
			32117	Palmerston Islander
			32118	Penrhyn Islander
			32119	Pukapuka Islander
			32120	Rakahanga Islander
			32121	Rarotongan
	33	Tongan		
		331	Tongan	
			33111	Tongan
	34	Niuean		
		341	Niuean	
			34111	Niuean
	35	Tokelauan		
		351	Tokelauan	
			35111	Tokelauan
	36	Fijian		
		361	Fijian	
			36111	Fijian
	37	Other Pacific Peoples		
		371	Other Pacific Peoples	
			37111	Admiralty Islander
			37112	Australian Aboriginal
			37113	Austral Islander
			37114	Palau Islander
			37115	Bismark Archipelagoan
			37116	Bougainvillean
			37117	Caroline Islander
			37118	Easter Islander
			37119	Gambier Islander
			37120	Guadalcanalian
			37121	Chamorro
			37122	Hawaiian
			37123	Kanak

			37124	Kiribati
			37125	Malaitian
			37126	Manus Islander
			37127	Marianas Islander
			37128	Marquesas Islander
			37129	Marshall Islander
			37130	Nauruan
			37131	New Britain Islander
			37132	New Georgian
			37133	New Irelander
			37134	Banaban
			37135	Papua New Guinean
			37136	Phoenix Islander
			37137	Pitcairn Islander
			37138	Rotuman
			37139	Santa Cruz Islander
			37140	Tahitian
			37141	Solomon Islander
			37142	Torres Strait Islander
			37143	Tuamotu Islander
			37144	Tuvaluan
			37145	Ni Vanuatu
			37146	Wake Islander
			37147	Wallis Islander
			37148	Yap Islander
			37199	Pacific Peoples nec
4	**Asian**			
	40	Asian nfd		
		400	Asian nfd	
			40000	Asian nfd
	41	Southeast Asian		
		410	Southeast Asian nfd	
			41000	Southeast Asian nfd
		411	Filipino	
			41111	Filipino
		412	Cambodian	
			41211	Cambodian
		413	Vietnamese	
			41311	Vietnamese
		414	Other Southeast Asian	
			41411	Burmese
			41412	Indonesian
			41413	Laotian
			41414	Malay
			41415	Thai
			41499	Southeast Asian nec
	42	Chinese		
		421	Chinese	

			42100	Chinese nfd
			42111	Hong Kong Chinese
			42112	Cambodian Chinese
			42113	Malaysian Chinese
			42114	Singaporean Chinese
			42115	Vietnamese Chinese
			42116	Taiwanese
			42199	Chinese nec
	43	Indian		
		431	Indian	
			43100	Indian nfd
			43111	Bengali
			43112	Fijian Indian
			43113	Gujarati
			43114	Indian Tamil
			43115	Punjabi
			43116	Sikh
			43117	Anglo Indian
			43199	Indian nec
	44	Other Asian		
		441	Sri Lankan	
			44100	Sri Lankan nfd
			44111	Sinhalese
			44112	Sri Lankan Tamil
			44199	Sri Lankan nec
		442	Japanese	
			44211	Japanese
		443	Korean	
			44311	Korean
		444	Other Asian	
			44411	Afghani
			44412	Bangladeshi
			44413	Nepalese
			44414	Pakistani
			44415	Tibetan
			44416	Eurasian
			44499	Asian nec
5	**Middle Eastern/Latin American/African**			
	51	Middle Eastern		
		511	Middle Eastern	
			51100	Middle Eastern nfd
			51111	Algerian
			51112	Arab
			51113	Assyrian
			51114	Egyptian
			51115	Iranian/Persian
			51116	Iraqi
			51117	Israeli/Jewish
			51118	Jordanian

		51119	Kurd
		51120	Lebanese
		51121	Libyan
		51122	Moroccan
		51123	Omani
		51124	Palestinian
		51125	Syrian
		51126	Tunisian
		51127	Turkish
		51128	Yemeni
		51199	Middle Eastern nec
52	Latin American		
	521	Latin American	
		52100	Latin American nfd
		52111	Argentinian
		52112	Bolivian
		52113	Brazilian
		52114	Chilean
		52115	Colombian
		52116	Costa Rican
		52117	Latin American Creole
		52118	Ecuadorian
		52119	Guatemalan
		52120	Guyanese
		52121	Honduran
		52122	Malvinian
		52123	Mexican
		52124	Nicaraguan
		52125	Panamanian
		52126	Paraguayan
		52127	Peruvian
		52128	Puerto Rican
		52129	Uruguayan
		52130	Venezuelan
		52199	Latin American nec
53	African		
	531	African	
		53100	African nfd
		53112	United States Creole
		53113	Jamaican
		53114	Kenyan
		53115	Nigerian
		53116	African American
		53117	Ugandan
		53118	West Indian
		53119	Somali
		53120	Eritrean
		53121	Ethiopian
		53122	Ghanaian

	53199	African nec

Appendix H – Interview Guide

This appendix shows the interview guide used for this study.

Interview Guide

Title of Research Project: A Mixed Methods Investigation of Ethnic Diversity and Productivity in Software Development

Name of Researcher: Jules Congalton
PhD Candidate
School of Management
Massey University
Wellington, New Zealand

Interview Overview

An interview will be held with the project manager for each software project nominated by the organisation. The interview will effectively be split into two parts, the first part focussing on qualitative data gathering and the second part on quantitative data. The first part will include open questions about how productive the project manager believes the project was and what factors affected the productivity of the project. Also in the first part of the interview will be questions about the team's diversity and whether that affected the team's performance, including productivity. The second part of the interview will focus on populating the variables identified in the conceptual model, such as the length of the project, the amount of staff turnover in the team and how often the requirements changed.

From the interviews and the review of project documentation, data will be sought on each project for each of the variables identified in the conceptual model. The size of the project in function points will be determined by counting the function points delivered by either reviewing system documentation or examining the system from an end user perspective. The total hours of effort expended to deliver the system will be captured. Ideally this will have already been captured through the organisations' timesheet systems. The number of lines of code delivered will be counted using tools for counting lines of source code. If this is not feasible, then the latest industry average lines of code per function point for the development language will be used.

Introduction and overview

- Greetings and thank you agreeing to participate
- Confirm participant received information sheet and consent form
- Provide a business card
- Provide overview of the research – including the rationale for the research project.
- Are there any questions about the information received or the research project
- Explain the interview is split into three sections: project and interviewee info, open Qs, model based Q's.
- State the preference to record interview - check participant(s) are okay with that.
- Okay to proceed?

- Sign consent form.
- Start recording
- Start interview

Section 0. Information About Interviewee

This first set of questions capture some basic information about the interviewee.

0.1. Name (pseudonym)

0.2. Gender (Circle one)

Male Female

0.3. Ethnicity

Select from Stats NZ Ethnicity list – use lowest level possible.

0.4. Number of years managing projects

0.5. Age group (Circle one)

Under 20 20 – 29 30-39 40-49 50-59 60-69 70+

Section 0A. Information About Organisation

This set of questions captures some basic information about the Organisation.

0.6. Name (Pseudonym)

0.7. Number of employees in organisation

0.8. Is software production the primary activity?

0.9. Government agency, Government owned or Private organisation?

Section 1. Project Summary Questions

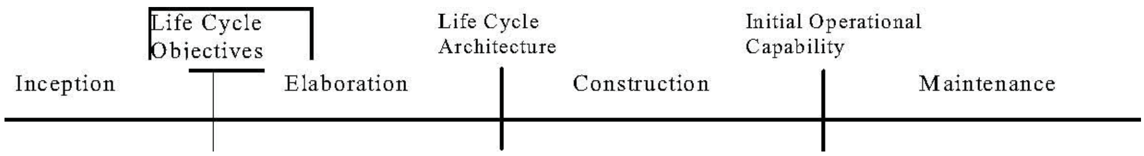
This set of questions captures basic information about the project. This includes things like application type and development activity being reported.

1.1 Project Name:

1.2 Month started and date completed: ____________ to: ________

1.3 Schedule Months. For reporting of historical data, provide the number of calendar months from the time the development began through the time it completed. For periodic reporting, provide the number of months in this development activity. Schedule in months:

Circle the life-cycle phases that the schedule covers:



1.4 Summary of the project purpose and deliverable(s)

1.5 Application Type. This field captures a broad description of the type of activity this software application is attempting to perform.

Circle One: Command and Control, MIS, Communication, Operating Systems, Diagnostics, Process Control, Engineering Signal processing, and Science. **Other:**

1.6 Development Type. Is the development a new software product or an upgrade of an existing

Circle One: New Product / Upgrade

1.7 Development Process. This is a description of the software process used to control the software development, for example, waterfall, spiral, etc.

1.8 Success Rating for Project. This specifies the degree of success for the project.

- Very successful; did almost everything right
- Successful; did the big things right
- OK; stayed out of trouble

- Some Problems; took some effort to keep viable
- Major Problems; would not do this project again

Circle One: Very Successful Successful OK Some Problems Major Problems

Section 2. Open Questions About Project

Open questions regarding productivity and team diversity

These next two questions relate to the productivity of the project.
Productivity is defined as the ratio of inputs to outputs and for this research project is defined as average hours to deliver each function point (a standard way of measuring software functionality), or lines of code per hour.
In other words, the amount of software delivered to the customer, for the amount of effort spent on the project.

2.1 Would you say it was a productive project team?

2.2 What factors affected the productivity of the project.

The next three questions relate to diversity within the team.
Diversity can be defined in many ways, including ethnicity, nationality, gender, age, experience, attitudes and communication style.
While the focus of this research is on ethnic diversity in the team, the next question relates to team diversity very broadly and is about any kind of any diversity at all that existed within the team.

2.3 Was there diversity within the project team? If so, in what ways was it diverse

2.4 What was the mix of ethnicities in the team (that is, how many people of each ethnicity were there).

The next three questions relate to the effects of ethnic diversity within the team.

2.5 Did the team's ethnic diversity affect the team's performance, including productivity?

2.7 What measures were taken to either mitigate the negative effect of ethnic diversity or enhance the positive effects of ethnic diversity?

2.8 How do you think the organizational context affected whether ethnic diversity had a positive or negative effect?

Section 3. COCOMO II Variables

This last section is a series of very specific closed questions based on a model of software development productivity called COCOMOII. There are 22 questions in this last section, but they are all short answer questions.

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
Precedentedness	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	Thoroughly familiar
Development Flexibility	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	General goals
Architecture / risk resolution[1]	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
Team cohesion	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions

Project Exponential Cost Drivers

Note 1: % significant risks eliminated

Enter the rating level for the first four cost drivers.

3.1 Precedentedness (PREC). If the project is similar to several that have been done before then the precedentedness is high. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.2 Development Flexibility (FLEX). This cost driver captures the amount of constraints the project has to meet. The more flexible the requirements, schedules, interfaces, etc., the higher the rating. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.3 Architecture / Risk Resolution (RESL). This cost driver captures the thoroughness of definition and freedom from risk of the software architecture used for the product. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.4 Team Cohesion (TEAM). The Team Cohesion cost driver accounts for the sources of project turbulence and extra effort due to difficulties in synchronizing the project's stakeholders: users, customers, developers, maintainers, interfacers, others. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

Effort

3.5 Total Effort (hours). Circle the life-cycle phases that the effort estimate covers:

Inception / Elaboration / Construction / Maintenance

Total Effort:

Size

The project would like to collect size in application points, logical lines of code, and unadjusted function points. Please submit all size measures that are available, for example, if you have a component in lines of code and unadjusted function points then submit both numbers.

3.6 Unadjusted Function Points. What is the total Unadjusted Function Points for each type. An Unadjusted Function Point is the product of the function point count and the weight for that type of point. Function Points are discussed in the Model Definition Manual.

3.7 Programming Language. What was the development language name that was used in this component, for example, Ada, C, C��, COBOL, FORTRAN and the amount of usage if more than one language was used.

Language	Percentage used

Product Cost Drivers.

For maintenance projects, identify any differences between the base code and modified code Product Cost Drivers (for example, complexity).

	Very Low	Low	Nominal	High	Very High	Extra High
RELY	slight inconvenience	Low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	
DATA		DB bytes/Pgm SLOC < 10	$10 \leq$ D/P < 100	$100 \leq$ D/P < 1000	D/P $\geq$ 1000	
RUSE		None	across project	across program	across product line	across multiple product lines
DOCU	Many life-cycle needs uncovered	Some life-cycle needs uncovered.	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	

3.8 Required Software Reliability (RELY). This is the measure of the extent to which the software must perform its intended function over a period of time. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.9 Develop for Reuse (RUSE). This cost driver accounts for the additional effort needed to construct components intended for reuse on the current or future projects. See the Model Definition Manual for more details.

Low	Nominal	High	Very High	Don't Know

3.10 Documentation match to life-cycle needs (DOCU). This captures the suitability of the project's documentation to its life-cycle needs. Several software cost models have a cost driver for the level of required documentation. In COCOMO II, the rating scale for the DOCU cost driver is evaluated in terms of the suitability of the project's documentation to its life-cycle needs.

Very Low	Low	Nominal	High	Very High	Don't Know

3.11 Product Complexity (CPLX):

	Control Operations	Computational Operations	Device-dependent Operations	Data Management Operations	User Interface Management Operations
Very Low	Straight-line code with a few non-nested structured programming operators: DOs, CASEs, IFTHENELSEs. Simple module composition via procedure calls or simple scripts.	Evaluation of simple expressions: e.g., *A=B+C*(D-E)*	Simple read, write statements with simple formats.	Simple arrays in main memory. Simple COTS-DB queries, updates.	Simple input forms, report generators.
Low	Straightforward nesting of structured programming operators. Mostly simple predicates	Evaluation of moderate-level expressions: e.g., *D=SQRT(B**2-4.*A*C)*	No cognizance needed of particular processor or I/O device characteristics. I/O done at GET/PUT level.	Single file subsetting with no data structure changes, no edits, no intermediate files. Moderately complex COTS-DB queries, updates.	Use of simple graphic user interface (GUI) builders.
Nom-inal	Mostly simple nesting. Some intermodule control. Decision tables. Simple callbacks or message passing, including middleware-supported distributed processing	Use of standard math and statistical routines. Basic matrix/vector operations.	I/O processing includes device selection, status checking and error processing.	Multi-file input and single file output. Simple structural changes, simple edits. Complex COTS-DB queries, updates.	Simple use of widget set.
High	Highly nested structured programming operators with many compound predicates. Queue and stack control. Homogeneous, distributed processing. Single processor soft real-time control.	Basic numerical analysis: multivariate interpolation, ordinary differential equations. Basic truncation, roundoff concerns.	Operations at physical I/O level (physical storage address translations; seeks, reads, etc.). Optimized I/O overlap.	Simple triggers activated by data stream contents. Complex data restructuring.	Widget set development and extension. Simple voice I/O, multimedia.
Very High	Reentrant and recursive coding. Fixed-priority interrupt handling. Task synchronization, complex callbacks, heterogeneous distributed processing. Single-processor hard real-time control.	Difficult but structured numerical analysis: near-singular matrix equations, partial differential equations. Simple parallelization.	Routines for interrupt diagnosis, servicing, masking. Communication line handling. Performance-intensive embedded systems.	Distributed database coordination. Complex triggers. Search optimization.	Moderately complex 2D/3D, dynamic graphics, multimedia.
Extra High	Multiple resource scheduling with dynamically changing priorities. Microcode-level control. Distributed hard real-time control.	Difficult and unstructured numerical analysis: highly accurate analysis of noisy, stochastic data. Complex parallelization.	Device timing-dependent coding, micro-programmed operations. Performance-critical embedded systems.	Highly coupled, dynamic relational and object structures. Natural language data management.	Complex multimedia, virtual reality.

Complexity is divided into five areas: control operations, computational operations, device-dependent operations, data management operations, and user interface management operations. Select the area or combination of areas that characterize the product or a sub-system of the product. The complexity rating is the subjective weighted average of these areas.

Very Low	Low	Nominal	High	Very High	Don't Know

Platform Cost Drivers. The platform refers to the target-machine complex of hardware and infrastructure software.

	Very Low	Low	Nominal	High	Very High	Extra High
TIME			= 50% use of available execution time	70%	85%	95%
STOR			=50% use of available storage	70%	85%	95%
PVOL		major change every 12 mo.; minor change every 1 mo.	major: 6 mo.; minor: 2 wk.	major: 2 mo.; minor: 1 wk.	major: 2 wk.; minor: 2 days	

3.12 Platform Volatility (PVOL). "Platform" is used here to mean the complex of hardware and software (OS, DBMS, etc.) the software product calls on to perform its tasks. See the Model Definition Manual for more details.

Low	Nominal	High	Very High	Don't Know

Personnel Cost Drivers.

	Very Low	Low	Nominal	High	Very High
ACAP	≤15th percentile	35th percentile	55th percentile	75th percentile	≥90th percentile
PCAP	≤15th percentile	35th percentile	55th percentile	75th percentile	≥90th percentile
PCON	≥48% / year	24% / year	12% / year	6% / year	≤3% / year
APEX	≤ 2 months	6 months	1 year	3 years	≥6 years
PEXP	≤ 2 months	6 months	1 year	3 years	≥6 years
LTEX	≤ 2 months	6 months	1 year	3 years	≥6 years

3.13 Analyst Capability (ACAP). Analysts are personnel that work on requirements, high level design and detailed design. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.14 Programmer Capability (PCAP). Evaluation should be based on the capability of the programmers as a team rather than as individuals. Major factors which should be considered in the rating are ability, efficiency and thoroughness, and the ability to communicate and cooperate. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.15 Applications Experience (APEX). This rating is dependent on the level of applications experience of the project team developing the software system or subsystem. The ratings are defined in terms of the project team's equivalent level of experience with this type of application. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.16 Platform Experience (PEXP). The Post-Architecture model broadens the productivity influence of PEXP, recognizing the importance of understanding the use of more powerful platforms, including more graphic user interface, database, networking, and distributed middleware capabilities. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.17 Language and Tool Experience (LTEX). This is a measure of the level of programming language and software tool experience of the project team developing the software system or subsystem. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.18 Personnel Continuity (PCON). The rating scale for PCON is in terms of the project's annual personnel turnover. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

Project Cost Drivers. This table gives a summary of the criteria used to select a rating level for project cost drivers.

	Very Low	Low	Nominal	High	Very High	Extra High
TOOL	edit, code, debug	simple, frontend, backend CASE, little integration	basic lifecycle tools, moderately integrated	strong, mature lifecycle tools, moderately integrated	strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse	
SITE: Collocation	International	Multi-city and Multi-company	Multi-city or Multi-company	Same city or metro area	Same building or complex	Fully collocated
SITE: Communications	Some phone, mail	Individual phone, FAX	Narrowband email	Wideband electronic communications	Wideband electronic communications, occasional video conferencing.	Interactive multimedia
SCED	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	

3.19 Use of Software Tools (TOOL). See the Model Definition Manual.

Very Low	Low	Nominal	High	Very High	Don't Know

3.20 Multisite Development (SITE). Given the increasing frequency of multisite developments, and indications that multisite development effects are significant, the SITE cost driver has been added in COCOMO II. Determining its cost driver rating involves the assessment and averaging of two factors: site collocation (from fully collocated to international distribution) and communication support (from surface mail and some phone access to full interactive multimedia). See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.21 Required Development Schedule (SCED). This rating measures the schedule constraint imposed on the project team developing the software. The ratings are defined in terms of the percentage of schedule stretch-out or acceleration with respect to a nominal schedule for a project requiring a given amount of effort. See the Model Definition Manual for more details.

Very Low	Low	Nominal	High	Very High	Don't Know

3.22 Process Maturity (PMAT). The procedure for determining PMAT is organized around the Software Engineering Institute's Capability Maturity Model (CMM). The time period for reporting process maturity is at the time the project was underway. We are interested in the capabilities practiced at the project level more than the overall organization's capabilities. "Overall Maturity Level" is a response that captures the result of an organized evaluation based on the CMM. "No Response" means you do not know or will not report the process maturity either at the Capability Maturity Model or Key Process Area level.

Overall Maturity Level

- CMM Level 1 (lower half)
- CMM Level 1 (upper half)
- CMM Level 2
- CMM Level 3
- CMM Level 4
- CMM Level 5

Appendix I – Data for All Variables for All Projects

Table 25. Raw data for projects 1 – 6

Variable	Project 01	Project 02	Project 03	Project 04	Project 05	Project 06
TMSZ	3	3			4	9
BEDI	0.67	0.67			0.67	0.56
NPDR	2.93	6.51			7.29	12.02
ETH	2	2			2	2
NHRS	117.13	527.09			349.93	119.02
UFP	40	81	0	0	48	99
PLNG	4	4	4	7	9	9
PMYR	18	18	18	18	18	18
PMAG	50-59	50-59	50-59	50-59	50-59	50-59
Govt	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
PMGN	Male	Male	Male	Male	Male	Male
PMET	European	European	European	European	European	European
REVL	Low	High			Nominal	High
AGIL	None	None			Some	Some
TYPE	Enhancement	Enhancement	Enhancement	Enhancement	Enhancement	Enhancement
SWPR	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
LANG	.NET	.NET			Ruby on Rails	Java
DVPN	Positive	Positive	Neutral	Positive	Neutral	Neutral
DIVT	Some	Some	Some	Some	Some	Some
PRPR	Very high	High	Low	High	High	Very low
PRSC	Successful	Successful	Major problems	Successful	Successful	Major problems
PMAT	Nominal	Nominal			Very low	Very low
CPLX	Nominal	High			Low	Nominal
SCED	Nominal	Nominal			Nominal	Nominal
SITE	Very high	Extra high			High	High
TOOL	Low	Low			Low	Low
PEXP	Very high	High			High	High
APEX	Very high	High			High	Nominal
PCAP	Very high	Very high			Nominal	Nominal
PVOL	Low	Low			Low	Low
DOCU	Nominal	Low			Low	Low
RUSE	Low	Low			Low	Low
RELY	Low	High			Low	Nominal
TEAM	High	High			High	Low
RESL	High	High			High	Very low
FLEX	Low	Nominal			Nominal	Low
PREC	Very high	Very high			High	High
PCON	Low	High			Very high	Low
LTEX	Very high	High			Low	Nominal

ACAP	Low	Low			High	Nominal

Table 26. Raw data for projects 7 – 13

Variable	Project 07	Project 08	Project 09	Project 10	Project 11	Project 12	Project 13
TMSZ			3	4	27	4	7
BEDI			0.67	0.5	0.48	0	0.81
NPDR	8.26	10.65	6.26	6.41	54.15	1.6	39.55
ETH			2	2	4	2	4
NHRS	231.33	1054.17	744.61	372	17815	176	13448.28
UFP	28	99	119	58	329	110	340
PLNG	4	4	6	7	21	5	14
PMYR	15	15	10	6	8	8	15
PMAG	40-49	40-49	40-49	40-49	40-49	30-39	60-69
Govt	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
PMGN	Female	Female	Male	Male	Male	Male	Male
PMET	European	European	European	Asian	European	European	European
REVL			Low	Low	High	Nominal	Low
AGIL			None	None	None	Some	Some
TYPE	Enhancement	Enhancement	New Development	Enhancement	Enhancement	Enhancement	New Development
SWPR	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
LANG	Java	Java	.NET	RPG	Pearl script	.NET	Java
DVPN			Neutral	Positive	Positive	Neutral	Neutral
DIVT			Some	Some	Some	None	Lots
PRPR	High	High	High	High	Low	High	Very high
PRSC			Successful	Successful	Successful	Some problems	Very successful
PMAT			High	Nominal	Nominal	Nominal	High
CPLX			Low	Low	Very high	Low	High
SCED			Nominal	Nominal	Nominal	Low	Nominal
SITE			High	Nominal	Low	Low	Very high
TOOL			Low	Nominal	Nominal	Nominal	Very high
PEXP			High	Very high	High	High	Nominal
APEX			Nominal	Very high	High	Very low	Very high
PCAP			Very high	High	High	Nominal	Very high
PVOL			Low	Nominal	Low	Very low	Low
DOCU	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal
RUSE	Extra high	Extra high	Very high	Nominal	Nominal	High	Extra high
RELY	High	High	High	Low	High	Low	Nominal
TEAM	Nominal	Nominal	High	Very high	Very high	Very high	Very high
RESL	High	High	Very high	High	Very high	Low	Low
FLEX	Very low	Very low	Low	High	Nominal	Very high	Nominal
PREC	Very high	Very high	Nominal	Extra high	High	Low	Very low

PCON	Very high	Very high	Low	Very high	Very high
LTEX	Very high	Very high	High	Nominal	Very high
ACAP	High	High	Very high	High	High

Table 27. Raw data for projects 13 – 19

Variable	Project 13	Project 14	Project 15	Project 16	Project 17	Project 18	Project 19
TMSZ	7	16	10	12	12	22	11
BEDI	0.81	0.63	0.53	0.55	0.41	0.39	0.65
NPDR	39.55	36.23	23.82	40.87	11.72	42.83	
ETH	4	4	3	3	3	3	3
NHRS	13448.28	6377.16	3858.67	2166.27	19109.91	21330.82	
UFP	340	176	162	53	1631	498	
PLNG	14	24	11	12	19	18	18
PMYR	15	9	25	25	15	20	20
PMAG	60-69	50-59	40-49	40-49	30-39	50-59	60-69
Govt	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
PMGN	Male	Female	Male	Male	Male	Male	Female
PMET	European	European	Asian	Asian	European	European	European
REVL	Low	High	Very high	High	High	High	Low
AGIL	Some	None	None	Some	Extensive	Some	None
TYPE	New Development	Enhancement	New Development	New Development	New Development	Enhancement	New Development
SWPR	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
LANG	Java	Java	Java	Java	Java	.NET	
DVPN	Neutral	Negative	Negative	Positive	Positive	Positive	Negative
DIVT	Lots	Lots	Some	Lots	Lots	Lots	Lots
PRPR	Very high	Low	Low	Average	High	High	High
PRSC	Very successful	Successful	Major problems	Successful	Very successful	Very successful	Very successful
PMAT	High	Low	Nominal	Nominal	Low	High	Low
CPLX	High	High	High	High	High	High	High
SCED	Nominal	Low	Nominal	Nominal	Nominal	Nominal	Nominal
SITE	Very high	Extra high	Very high	Very high	Extra high	Very high	Very high
TOOL	Very high	Low	Nominal	Nominal	High	High	Very high
PEXP	Nominal	Nominal	High	Nominal	Very low	High	High
APEX	Very high	Low	Very low	Nominal	High	Nominal	Nominal
PCAP	Very high	Nominal	High	High	High	Very high	High
PVOL	Low	Very high	Nominal	Nominal	Low	Nominal	Low
DOCU	Nominal	High	Low	Low	Nominal	Nominal	High
RUSE	Extra high	Extra high	Extra high	Extra high	Extra high	Very high	Very high
RELY	Nominal	High	High	High	Low	Nominal	High

TEAM	Very high	Low	Very low	Very high	Nominal	High	High
RESL	Low	Very low	Low	Very high	High	Very high	Very high
FLEX	Nominal	Low	Very low	Very low	Nominal	Low	Low
PREC	Very low	Nominal	Extra high	Extra high	Low	Very low	Low
PCON	Very high	Very low	Very high	Very high	Very high	High	Nominal
LTEX	Very high	High	High	High	Very high	High	Very high
ACAP	High	Nominal	High	Very high	High	High	High

Appendix J – Coding Frequency of Productivity Factors

Table 28 is an extract from NVivo, the qualitative data analysis software tool used, which shows the number of interviews where each theme regarding what influenced productivity was discussed by the project managers.

Table 28. The number of interviews where each productivity factors is discussed

Name	Sources
Influences of productivity	0
Communication	14
Team cohesion	11
Requirements quality and volatility	10
Team capability	9
Attitude and motivation	7
Agile practices	7
Complexity	4
Application experience	4
Planning and design	4
Resource availability	4
Dedicated project team	3
Scope Management	2
Platform experience	2
Managing expectations	1
Organisational change	1
Analyst capability	1
Co-located project team	1

Appendix K – Use of "We" or "They" to Refer to Software Team

Table 29 shows whether "we", "they" or "the team" was predominantly used by the project manager in response to the interview questions about productivity. The first question was "was the project team productive?" and the second is "what affected the productivity of the project team?".

Table 29. Use of "we" or "they" to refer to software team

Org	Project	Was it prod?	What affected prod?	Overall we / they	Govt?	Compared to sector mean (low is more productive)
D	10	we	we	we	Y	20%
E	17	we	we	we	Y	36%
A	1	we	we	we	N	42%
E	15	the team	we	mixed	Y	73%
C	9	they	we	mixed	N	89%
A	2	we	we	we	N	93%
A	5	the team	we	mixed	N	104%
E	14	we	we	we	Y	110%
D	12	we	we	we	Y	121%
E	13	we	the team	mixed	Y	121%
E	16	we	the team	mixed	Y	125%
F	18	the team	we	mixed	Y	131%
D	11	the team	we	mixed	Y	165%
A	6	we	we	we	N	172%

Appendix L – Kendall Rank-Order Correlations

The following table show the Kendall rank-order correlation coefficients for all ethnic diversity variables and productivity. All correlations reported are for a two tailed test.

Table 30. The Kendall rank-order correlations

		Blau's index of ethnic diversity	Number of ethnicities	Normalised Project Delivery Rate (Productivity)
Blau's index of gender diversity	Correlation Coefficient	.040	.035	.069
	Sig. (2-tailed)	.841	.871	.727
	N	15	15	15
Team size	Correlation Coefficient	-.396*	.683**	.709**
	Sig. (2-tailed)	.049	.002	.001
	N	15	15	14
Percentage of female	Correlation Coefficient	.020	.325	.311
	Sig. (2-tailed)	.919	.141	.133
	N	15	15	14
Normalised Project Delivery Rate (Productivity)	Correlation Coefficient	-.125	.685**	1.000
	Sig. (2-tailed)	.542	.002	.
	N	14	14	16
Normalised hours	Correlation Coefficient	-.262	.658**	.550**
	Sig. (2-tailed)	.202	.003	.003
	N	14	14	16
Blau's index of ethnic diversity	Correlation Coefficient	1.000	-.107	-.125
	Sig. (2-tailed)	.	.624	.542
	N	15	15	14
Unadjusted function points	Correlation Coefficient	-.284	.523*	.377*
	Sig. (2-tailed)	.166	.019	.043
	N	14	14	16
Number of ethnicities	Correlation Coefficient	-.107	1.000	.685**
	Sig. (2-tailed)	.624	.	.002
	N	15	15	14
Project Length in months	Correlation Coefficient	-.189	.764**	.593**
	Sig. (2-tailed)	.340	.000	.002
	N	15	15	16
Years managing projects	Correlation Coefficient	.062	.061	.195
	Sig. (2-tailed)	.761	.785	.312
	N	15	15	16
Project Manager Age	Correlation Coefficient	.580**	.134	.170
	Sig. (2-tailed)	.007	.568	.407
	N	15	15	16
Sector	Correlation Coefficient	-.107	.698**	.542*

		Blau's index of ethnic diversity	Number of ethnicities	Normalised Project Delivery Rate (Productivity)
	Sig. (2-tailed)	.640	.006	.013
	N	15	15	16
Project Manager Gender	Correlation Coefficient	.118	.372	.044
	Sig. (2-tailed)	.607	.142	.840
	N	15	15	16
Project Manager Ethnicity	Correlation Coefficient	-.201	.000	.102
	Sig. (2-tailed)	.382	1.000	.638
	N	15	15	16
Requirements Evolution and Volatility (REVL)	Correlation Coefficient	-.353	.278	.436*
	Sig. (2-tailed)	.104	.244	.048
	N	15	15	14
Use of Agile Practices (AGIL)	Correlation Coefficient	-.217	.030	.127
	Sig. (2-tailed)	.333	.903	.577
	N	15	15	14
Enhancement or New Development	Correlation Coefficient	.150	.323	.185
	Sig. (2-tailed)	.513	.203	.396
	N	15	15	16
Is software production the primary activity	Correlation Coefficient	.409	-.573*	-.306
	Sig. (2-tailed)	.075	.024	.159
	N	15	15	16
Process Maturity (PMAT)	Correlation Coefficient	.000	.108	.065
	Sig. (2-tailed)	1.000	.646	.767
	N	15	15	14
Required Development Schedule (SCED)	Correlation Coefficient	.237	-.093	.171
	Sig. (2-tailed)	.304	.714	.465
	N	15	15	14
Multisite Development (SITE)	Correlation Coefficient	.221	.313	.146
	Sig. (2-tailed)	.295	.177	.495
	N	15	15	14
Use of Software Tools (TOOL)	Correlation Coefficient	-.341	.497*	.325
	Sig. (2-tailed)	.111	.035	.140
	N	15	15	14
Platform Experience (PLEX)	Correlation Coefficient	.025	-.559*	-.368
	Sig. (2-tailed)	.909	.020	.096
	N	15	15	14
Applications Experience (APEX)	Correlation Coefficient	.349	-.090	-.085
	Sig. (2-tailed)	.097	.696	.692
	N	15	15	14
Programmer Capability (PCAP)	Correlation Coefficient	.245	.028	.013
	Sig. (2-tailed)	.260	.908	.953
	N	15	15	14
Platform volatility	Correlation Coefficient	-.114	.359	.403

		Blau's index of ethnic diversity	Number of ethnicities	Normalised Project Delivery Rate (Productivity)
(PVOL)	Sig. (2-tailed)	.604	.139	.072
	N	15	15	14
Product Complexity (CPLX)	Correlation Coefficient	-.108	.766**	.674**
	Sig. (2-tailed)	.619	.001	.002
	N	15	15	14
Documentation Match to Life-Cycle Needs (DOCU)	Correlation Coefficient	-.099	.351	-.034
	Sig. (2-tailed)	.654	.150	.873
	N	15	15	16
Developed for Reusability (RUSE)	Correlation Coefficient	-.153	.618**	.258
	Sig. (2-tailed)	.467	.007	.197
	N	15	15	16
Required Software Reliability (RELY)	Correlation Coefficient	.155	.394	.318
	Sig. (2-tailed)	.479	.102	.125
	N	15	15	16
Team Cohesion (TEAM)	Correlation Coefficient	-.011	.000	-.149
	Sig. (2-tailed)	.958	1.000	.453
	N	15	15	16
Architecture / Risk Resolution (RESL)	Correlation Coefficient	-.089	.053	.068
	Sig. (2-tailed)	.675	.822	.737
	N	15	15	16
Development Flexibility (FLEX)	Correlation Coefficient	-.057	-.187	-.226
	Sig. (2-tailed)	.791	.425	.259
	N	15	15	16
Precedentedness (PREC)	Correlation Coefficient	.042	-.258	-.116
	Sig. (2-tailed)	.839	.250	.551
	N	15	15	16
Personnel Continuity (PCON)	Correlation Coefficient	-.047	-.181	-.205
	Sig. (2-tailed)	.826	.444	.356
	N	15	15	14
Language and Tool Experience (LTEX)	Correlation Coefficient	.199	.152	-.091
	Sig. (2-tailed)	.359	.524	.680
	N	15	15	14
Analyst Capability (ACAP)	Correlation Coefficient	-.322	.365	.379
	Sig. (2-tailed)	.138	.126	.086
	N	15	15	14

**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.05 level (2-tailed).

Appendix M – Concept Map of Interview Themes

The following diagram shows a concept map (Moon, et al., 2011; Nilson, 2009) of the relationships between the themes that were present in the interviews.

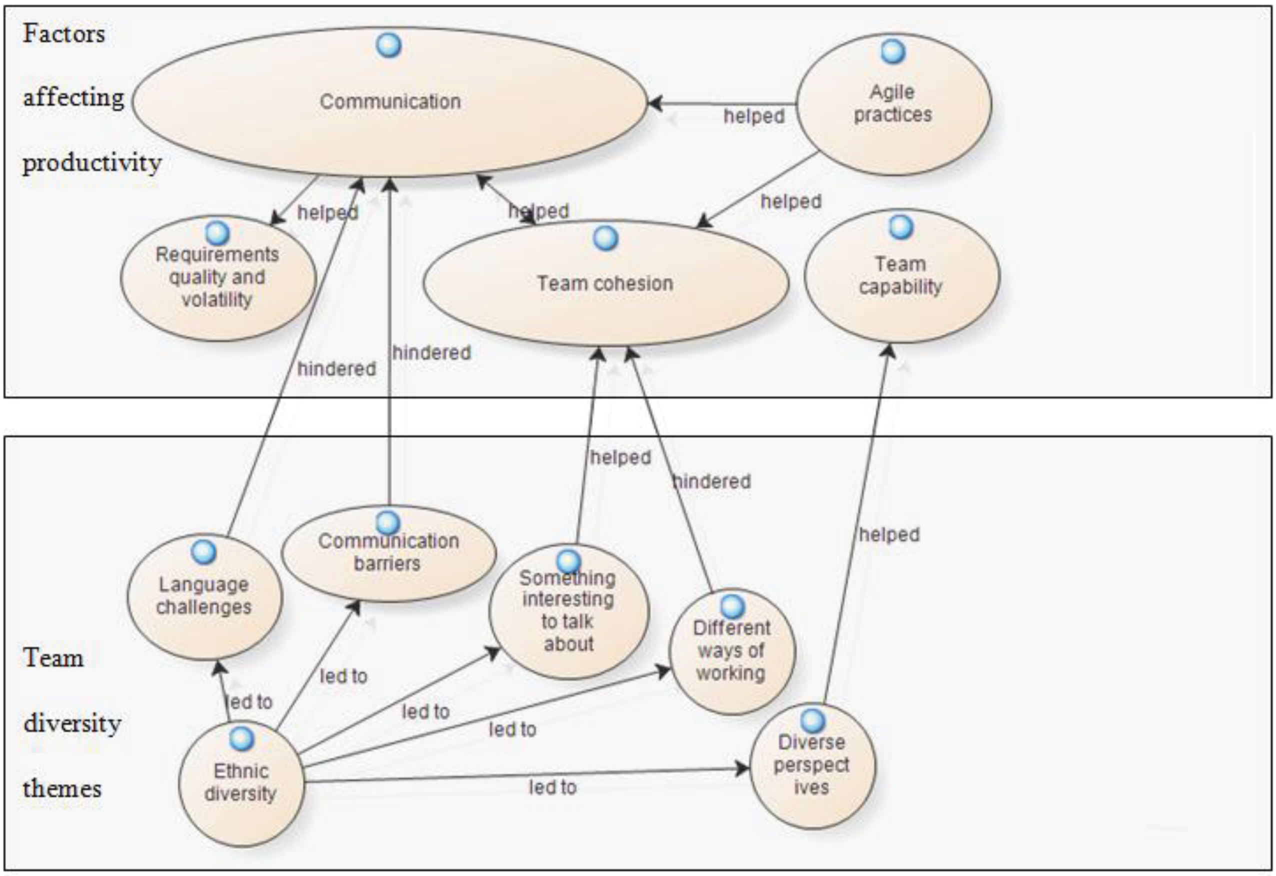


Figure 13. Concept map of major productivity factors and the role of ethnic diversity

.