GLOBALLY DISTRIBUTED AGILE TEAMS: AN EXPLORATORY STUDY OF THE DIMENSIONS CONTRIBUTING TO SUCCESSFUL TEAM CONFIGURATION

Jason H. Sharp, B.S., M.DIV., M.S.

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

December 2008

APPROVED:

Sherry D. Ryan, Committee Co-Chair
Victor R. Prybutok, Committee Co-Chair
Cathleen A. Norris, Committee Member
Jon I. Young, Committee Member
Linda Schamber, Associate Director of the Interdisciplinary Ph.D. Program
Herman L. Totten, Dean of the School of Information, Library Science, and Technologies
Sandra L. Terrell, Dean of the Robert B. Toulouse School of Graduate Studies

Sharp, Jason H. Globally Distributed Agile Teams: An Exploratory Study of the Dimensions Contributing to Successful Team Configuration. Doctor of Philosophy (Information Science), December 2008, 552 pp., 104 tables, 5 figures, references, 148 titles.

Drawing upon configurational theory, work group design research, virtualness concepts, and the software agility literature, the purpose of this study was to provide a starting point for theorizing about the successful configuration of globally distributed agile teams by exploring the dimensions of team structure, virtualness, and agility.

Due to the complex nature of this topic, the need to examine the phenomenon within its natural setting, and the limited amount of research that has been conducted in this particular area, this study adopted an embedded multiple-case research design. The primary data collection method consisted of semi-structured interviews involving members of globally distributed agile teams within three U.S. based organizations with members located in distributed sights in multiple countries. Additional data were collected from archival records. Within-case and cross-analysis was conducted using qualitative data analysis software.

This study provides a starting point for answering the question of how the configuration of globally distributed agile teams differs from the configuration of other types of globally distributed teams; it synthesizes past research and findings into a comprehensive theoretical framework; it provides a starting point for theorizing about the successful configuration of globally distributed agile teams; it helps practitioners to identify and address the challenges related to the configuration of globally distributed agile teams; and it presents a set of best practices which will inform organizations on how to configure their globally distributed agile teams.

ACKNOWLEDGEMENTS

I would like to thank my committee co-chairs, Dr. Sherry Ryan and Dr. Victor Prybutok, for their consistent and timely feedback throughout the dissertation process. In particular, I would like to express my sincere gratitude to Dr. Ryan for taking an interest in my research early on and encouraging me to take advantage of numerous opportunities that I otherwise would have been unaware of and that have proved to be invaluable. I would also like to thank my committee members, Dr. Cathleen Norris and Dr. Jon Young, for their guidance early on in my program and their support throughout.

Personally, I would like to thank my parents, James and Billie Sharp, for their unending support and for providing whatever I needed to accomplish this goal. I would like to also thank my friend and mentor these many years, Dan Rivers, for the steadfast counsel and camaraderie. To my peer support group who spurned me on I say a huge “thank you”: Dr. Jason Huett, Danny, Mendy, and Jana Golightly, Leah Schultz, Mark Adams, Trissa Cox, Dr. Bashorat Ibragimova, and Jason Berryhill.

I would like to thank my many colleagues in the Department of Computer Information Systems and the College of Business Administration at Tarleton State University who have supported me throughout this process. Specifically: Dr. Ruby Barker, Dr. Brad Chilton, Dr. Dan Collins, Dr. Raja Iyer, Dr. Steve Steed, and Dr. Linda Jones. In addition, I would like to express my thanks to the administration at Tarleton State who supported me in so many ways during this endeavor.

I would like to thank the organizations and individuals who participated in this study for their willingness to give of their time and input.

I thank God for His great faithfulness even in the times when I lacked it myself.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS..iii

LIST TABLES..vi

LIST OF FIGURES..xi

Chapters

1. INTROD UCTION 1

Background 1

Problem Statement 2

Purpose of the Study 2

Research Question 3

Research Design 3

Significance of the Study 4

Organization of the Dissertation 5

2. LITERATURE REVIEW 6

Agile Software Development Methods 6

Global Software Development 19

Agile Methods and Globally Distributed Environments 42

Summary and Implications of Literature Review 57

Theoretical Framework 58

3. RESEARCH METHOD 78

The Case Study as a Research Approach 78

The Use of Case Study Research in Information Systems 79

The Use of Case Study Research to Build Theory........................ 80

4. DATA ANALYSIS... 112

Organization A Cases 112

A01 Within-Case Analysis.. 120

A02 Within-Case Analysis.. 221

Organization B Cases 250

B01 Within-Case Analysis.. 255

B02 Within-Case Analysis.. 292

Organization C Case.. 322

C01 Within-Case Analysis.. 328

Cross-Case Analysis.. 419

5. DISCUSSION AND CONCLUSIONS.. 481

Discussion of Research Findings.. 481

Limitations of the Study.. 501

Recommendations for Future Research 508

Appendices

A. INFORMED CONSENT DOCUMENTS..……………………………………513

B. INTERVIEW PROTOCOL…………………………………………………...516

C. CODING LIST…………………………………………………………………...525

D. CASE STUDY PROTOCOL…………………………………………………528

REFERENCES……………………………………………………………………………536

LIST OF TABLES

Page

1. Twelve Guiding Principles of Agile Software Development Methods……………..7

2. Brief Definitions of Theoretical Framework Dimensions…………………………..60

3. Challenges to Globally Distributed Agile Teams…………………………………...61

4. Process of Building Theory from Case Study Research…………………………..81

5. Case Study Protocol Steps…………………………………………………………...94

6. Within-Case Analysis Procedure…………………………………………………...103

7. Case Study Tactics Adopted for Design Quality…………………………………109

8. A01 Team Size, Selection, Locations, Project, and Duration……………………113

9. A01 Demographics for Members Who Participated in the Study……………….115

10. A01 Current Members and Geographic Locations………………………………117

11. A01 Team Structure Summary of Findings………………………………………..161

12. A01 Extent (Degree) Team Virtualness……………………………………………168

13. A01 Benefits and Challenges of Temporal Distribution………………………....176

14. A01 Team Virtualness Summary of Findings……………………………………..188

15. A01 Summary of Team Agility………………………………………………………195

16. A01 Familiarity with the Agile Manifesto…………………………………………...196

17. A01 Responses to Adherence to Values of the Agile Manifesto………………..197

18. A01 Extent (Degree) of Agility for Values…………………………………………198

19. A01 Responses to Adherence to Principles of the Agile Manifesto…………….200

20. A01 Extent (Degree) of Agility for Principles………………………………………203

21. A01 Extent (Degree) of Agility for Practices……………………………………….205

22. A01 Agile Methodology Implementation Summary……………………………….215

23. A02 Team Size, Selection, Locations, Project, and Duration……………………218

24. A02 Demographics for Members Who Participated in the Study……………….219

25. A02 Initial and Current Members and Geographic Locations…………………...220

26. A02 Team Structure Summary of Findings………………………………………..226

27. A02 Extent (Degree) of Team Virtualness…………………………………………228

28. A02 Benefits and Challenges of Temporal Distribution……………………….....232

29. A02 Team Virtualness Summary of Findings……………………………………..237

30. A02 Summary of Team Agility………………………………………………………240

31. A02 Extent (Degree) for Values…………………………………………………….241

32. A02 Extent (Degree) of Agility for Principles………………………………………243

33. A02 Extent (Degree) of Agility for Practices……………………………………….245

34. A02 Agile Methodology Implementation Summary……………………………….249

35. B01 Team Size, Selection, Locations, Project, and Duration……………………251

36. B01 Demographics for Members Who Participated in the Study……………….252

37. B01 Current Members and Geographic Locations……………………………….....253

38. B01 Team Structure Summary of Findings………………………………………..263

39. B01 Extent (Degree) of Team Virtualness…………………………………………268

40. B01 Benefits and Challenges of Temporal Distribution……………………….....271

41. B01 Team Virtualness Summary of Findings……………………………………..276

42. B01 Summary of Team Agility………………………………………………………279

43. B01 Extent (Degree) of Agility for Values………………………………………….281

44. B01 Extent (Degree) of Agility for Principles………………………………………283

45. B01 Extent (Degree) of Agility for Practices……………………………………….285

46. B01 Agile Methodology Implementation Summary……………………………….288

47. B02 Team Size, Selection, Locations, Project, and Duration……………………289

48. B02 Demographics for Members Who Participated in the Study……………….290

49. B02 Current Members and Geographic Locations...……………………………..291

50. B02 Team Structure Summary of Findings………………………………………..300

51. B02 Extent (Degree) of Team Virtualness…………………………………………303

52. B02 Benefits and Challenges of Temporal Distribution………………………....307

53. B02 Team Virtualness Summary of Findings……………………………………..312

54. B02 Summary of Team Agility……………………………………………………314

55. B02 Extent (Degree) of Agility for Values…………………………………………316

56. B02 Extent (Degree) of Agility for Practices……………………………………….318

57. B02 Agile Methodology Implementation Summary……………………………….321

58. C01 Team Size, Selection, Locations, Project, and Duration…………………...323

59. C01 Demographics for Members Who Participated in the Study……………….324

60. C01 Current Members and Geographic Locations……………………………….326

61. C01 Team Structure Summary of Findings………………………………………..352

62. C01 Extent (Degree) of Team Virtualness………………………………………...360

63. C01 Benefits and Challenges of Temporal Distribution………………………….368

64. C01 Team Virtualness Summary of Findings……………………………………..387

65. C01 Summary of Team Agility……………………………………………………395

66 C01 Familiarity with the Agile Manifesto…………………………………………..396

67. C01 Responses to Adherence to Values of the Agile Manifesto………………..397

68. C01 Extent (Degree) of Agility for Values……………………………………………398

69. C01 Extent (Degree) of Agility for Principles………………………………………404

70. C01 Extent (Degree) of Agility for Practices………………………………………407

71. C01 Agile Development Process…………………………………………………..413

72. C01 Agile Methodology Implementation Summary………………………………417

73. Structure - Task Design - Meaningfulness……………………..………………423

74. Structure - Task Design - Autonomy………………………………………………426

75. Structure - Task Design - Feedback………………………………………………429

76. Structure - Core Norms…………………………………………………………...433

77. Structure - Composition - Size - Small……………………………………………..437

78. Structure - Composition - Size - Large……………………………………………438

79. Structure - Composition - Mix……………………………………………………441

80. Structure - Composition - Knowledge and Task-Related Skills…………………445

81. Structure - Composition - Interpersonal Skills……………………………………448

82. Virtualness - Temporal Distribution - Time Zone Differences…………………...450

83. Virtualness - Temporal Distribution - Use of ICT…………………………………451

84. Virtualness - Boundary Spanning - Functional……………………………………454

85. Virtualness - Boundary Spanning - Organizational………………………………455

86. Virtualness - Boundary Spanning - Cultural………………………………………456

87. Virtualness - Lifecycle - Short……………………………………………………459

88. Virtualness - Lifecycle - Long……………………………………………………460

89. Virtualness - Member Roles - Within One Team…………………………………463

90. Virtualness - Member Roles - Within Multiple Teams……………………………464

91. Overall Familiarity with the Agile Manifesto and Adherence to Values and Principles………………..……… ………………………………………………...466
92. Overall Adherence to Values and Principles of the Agile Manifesto by Team…………………................ 467
93. Cross-Case Analysis Extent (Degree) of Agility for Principles…………………..468
94. Cross-Case Analysis Extent (Degree) of Agility for Practices…………………..470
95. Summary of Propositions related to Structure……………………………………472
96. Summary of Propositions related to Virtualness.…………………………………478
97. Summary of Propositions related to Agility.………………………………………479
98. Summary of Literature Related to Agility Dimension of Distributed Agile Teams…………………………………………………………………………………481
99. Summary of the Team Agility………………………………………………………484
100. Summary of Literature Related to Structure Dimension of Distributed Agile Teams…………………………………………………………………………………485
101. Summary of the Intersection of Agility and Structure……………………………493
102. Summary of Literature Related to Virtualness Dimension of Distributed Agile Teams…………………………………………………………………………………494
103. Summary of the Intersection of Agility and Virtualness…………………………500
104. Best Practices for a Successful Globally Distributed Agile Team Across Five Teams…………………………………………………………………………………507

LIST OF FIGURES

Page

1. Globally Distributed Agile Team Configuration Theoretical Framework…………64

2. Code Hierarchy in MAXQDA……………………..…………………………………100

3. Revised Theoretical Framework…………………………………………………505

4. Representation of Contingency Theory in MIS Research……………………….509

5. Proposed Theoretical Framework for Future Research Including Team

Processes………………………………………………………………………..5 11

CHAPTER 1

INTRODUCTION

Background

Projects that are over-budget, delivered late, and fall short of user's expectations have been a common problem area for software development efforts for years. Agile methods, which represent an emerging set of software development methodologies based on the concepts of adaptability and flexibility, are currently touted as a way to alleviate these reoccurring problems and pave the way for the future of development (Abrahamson, Warsta, Sippon, & Ronkainen, 2003; Augustine, Payne, Sencindiver, & Woodcock, 2005; Erickson, Lyytinen, & Siau, 2005). Empirical evidence does exists which appears to support the claims made by advocates of agile methods (Abrahamson, 2003; Abrahamson & Koskel, 2004). However, the adoption of agile methods entails both benefits and challenges in such areas as requirements determination, project management, and implementation.

Parallel interest is also growing in global software development and the formation of virtual teams. The benefits of drawing from a large group of talented software developers available around the world, a 24-hour work schedule, and the ability to rapidly form and deploy virtual teams is a strong lure to many organizations attempting to address the challenges faced in software development. However, this increasing movement toward global software development through the use of distributed virtual teams poses a potential dilemma for organizations who have adopted agile methods. A fundamental principle of agile methods is the efficacy of co-located teams in order to enable daily, face-to-face interaction between stakeholders (Fowler & Highsmith, 2001).

Within the context of global software development virtual teams must rely to varying degrees on technology-mediated communications and in some cases team members never actually meet in person (Gibson & Cohen, 2003).

Problem Statement

This raises the question of whether it is possible to successfully adopt agile methods for use in globally distributed environments. The perspective held by many of the early advocates of agile has been that these methods were not applicable to global software development projects (Holmstrom, Fitzgerald, Agerfalk, & Conchuir, 2006). A growing stream of research suggests that, although it is sometimes difficult and takes great care, it is possible (e.g., Kircher, Prashant, Corsaro, & Levine, 2001; Ramesh, Cao, Mohan, & Xu, 2006; Schummer & Schummer, 2001; Xiaohu, Bin, Zhijun, & Maddineni, 2004). The results of this research indicate that the key to successful implementation is to modify the agile method to fit the global setting. This concept of tailoring agile practices to fit the development context has been suggested by others as well (Fitzgerald, Hartnett, & Conboy, 2006). If this research holds true, the use of agile software development methods in globally distributed environments through the use of virtual teams represents a promising combination for creating a viable solution to the traditional challenges faced in software development.

Purpose of the Study

Team configuration in global settings is a complex phenomenon. According to Olson and Olson (2000), “collaborative work at a distance will be difficult to do for a long time, if not forever" (p. 163). While it is true that globally distributed teams encounter many of the same challenges as co-located teams, these are often exacerbated by physical distance and sometimes cultural issues (Komi-Sirvio & Tihinen, 2005). Thus, the actual configuration of agile teams in globally distributed environments appears to be a significant area of research that has currently received little attention. Therefore, the purpose of this study was to generate theory for exploring the dimensions contributing to the successful configuration of globally distributed agile teams that was grounded in the data and integrated with prior research.

Research Question

Drawing upon configurational theory, work group design research, team virtualness concepts, and the extant literature on agile methods and global software development, a preliminary theoretical framework was constructed for exploring the following research question: how can agile software development teams be successfully configured in globally distributed environments? The theoretical framework consists of three dimensions of team configuration: structure, virtualness, and agility. The framework also proposes that successful team configuration is impacted by multiple challenges associated with agile methods and global software development through the use of virtual teams.

Research Design

Due to the complex nature of this topic, the need to examine the phenomenon within its natural setting, and the limited amount of research that has been conducted in this particular area, this study adopted an embedded multiple-case research design (Benbasat, Goldstein, and Mead 1987; Bonoma 1985; Eisenhardt, 1989; Yin 2003). The primary data collection method consisted of semi-structured interviews involving individuals from five teams selected from three U.S. based organizations with members distributed across multiple countries. Additional data were collected via archival records including hierarchy charts and lists of roles and responsibilities. Within-case and cross-analysis was used to analyze the collected data using MAXQDA, a qualitative data analysis software package.

Significance of the Study

First, this study makes a contribution to academic research in the following ways: (1) it supports previous literature indicating that agile methods can be successfully implemented in globally distributed environments; (2) it provides a starting point for answering the question of how the configuration of globally distributed agile teams differs from the configuration of other types of globally distributed teams; (3) it synthesizes past research and the findings of the study into a comprehensive theoretical framework which is grounded in the data that can be utilized by researchers since no well-accepted framework currently exists; and (4) it provides a starting point for theorizing about how agile teams can be successfully configured in a globally distributed environment.

Second, this study makes a contribution to practice in the following ways: (1) it synthesizes past research and current findings into a theoretical framework which can also be beneficial to practitioners; (2) it helps practitioners to identify and address the challenges related to the configuration of globally distributed agile teams; (3) it presents a set of best practices which organizations can follow when configuring this specific type of team that is informed by both the data and the existing literature.

Organization of the Dissertation

This dissertation is organized as follows: Chapter 2 reviews the extant literature in the areas of agile software development methods, global software development, virtual teams, and the use of agile methods in globally distributed. Chapter 2 also presents the preliminary theoretical framework constructed for the study and provides a visual representation and verbal description of its dimensions. Chapter 3 describes the research method utilized for this study and outlines the case study protocol and interview protocol that were employed. Chapter 4 presents the findings of the within-case and cross-case analysis as well as the propositions put forth from interpretation of the data. Chapter 5 concludes with a discussion of the findings, limitations, contributions, and recommendations for future research.

CHAPTER 2

LITERATURE REVIEW

This chapter includes the major streams of literature necessary for establishing the foundation of the study. The first stream of literature relates to agile software development methods. The second stream reviews research in the areas of global software development and virtual teams. The third stream of literature examines the use of agile software development methods in globally distributed environments. Based upon these streams of literature and the elements of work group design research the theoretical framework for the study is developed. The chapter concludes with a summary and the implications of the literature review.

Agile Software Development Methods

With the volatile nature of business environments, rapidly changing requirements, emerging technologies, and the traditionally high rates of failure, the development of software in a timely and cost-effective manner which meet the needs of an organization continue to be a significant concern (e.g., Augustine et al., 2005; Erickson et al., 2005; Highsmith & Cockburn, 2001; Lycett, Macredie, Patel, & Paul, 2003; Reifer, 2002). One of the current proposed solutions to this challenge has been the creation of agile software development methods. Highsmith and Cockburn (2001) contend that agile methods “view change from a perspective that mirror today’s turbulent business and technology environment” (p. 120).

The term agile methods grew out of a meeting of scholars and practitioners in 2001 who were interested in establishing common ground among various development methodologies originating from the 1990s. The outcome of this meeting was a statement entitled the “Manifesto for Agile Software Development” which summarized the major tenets of the methodology as well as established a set of twelve guiding principles shown in Table 1 (Fowler & Highsmith, 2001). The Agile Manifesto states the following:

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. That is, while there is value in the items on the right, we value the items on the left more (Fowler & Highsmith, 2001).

Table 1

Twelve Guiding Principles of Agile Software Development Methods

# Principle	Description
1.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2.	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

(table continues)

Table 1 *(continued)*.

# Principle	Description
4.	Business people and developers must work together daily throughout the project.
5.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7.	Working software is the primary measure of progress.
8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9.	Continuous attention to technical excellence and good design enhances agility.
10.	Simplicity--the art of maximizing the amount of work not done--is essential.
11.	The best architectures, requirements, and designs emerge from self-organizing teams.
12.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Brief Historical Overview of Development Methods

When investigating an emergent development methodology it is often helpful to first construct a historical perspective on the subject in general. Avison and Fitzgerald (2003) provided a brief overview of the history of systems development methodologies. They identified three specific eras: pre-, early, and post-methodology.

In the pre-methodology era the focus was on programming and technical problem solving without utilizing a formal development method. The early methodology era saw the creation of the Systems Development Life Cycle (SDLC) or Waterfall method which centered on the completion of specific phases. Hoffer, George, and Valacich (2005) identified these phases as planning, analysis, design, implementation, and maintenance. Although the SDLC became the foundation for many development projects, it contains several limitations including failure to meet user needs, instability, inflexibility, user dissatisfaction, application backlog and high maintenance costs (Avison & Fitzgerald, 2003; Hoffer et al., 2005). Avison and Fitzgerald (2003) listed seven categories which emerged in the methodology era which included: structured, data-oriented, prototyping, object-oriented, participative, strategic, and systems. Finally, the authors characterized the post-methodology era as a time of “serious reappraisal by researchers and practitioners alike of the concepts and usefulness of the earlier methodologies” (Avison & Fitzgerald, p. 80).

According to the categories identified by Avison and Fitzgerald (2003), agile development methods fall into post-methodology era. For some organizations, “the problem is not the concept of a methodology, but the inadequacy of the current methodologies, prompting them to keep looking for different and better ones” (Avison &

Fitzgerald, p. 81). One type of development mentioned was incremental which is characterized by allowing the continuous enhancement of the system as well as addressing requirements during the development process. Both of these characteristics are shared by agile methods.

Overview of Specific Agile Methods

As Abrahamson et al. (2003) noted, agile methods do not explain simply one way of development, but rather a group of methods built on the idea of flexibility and adaptability. Consequently, nine distinct agile methods have been identified in the literature (Abrahamson et al., 2003; Meso & Jain, 2006). A brief description for each of these methods is provided below.

Adaptive Software Development (ASD) (Highsmith, 1999) stresses incremental and iterative development with continuous prototyping. Its emphasis is on the production of high-value results derived from rapid adaptation to both external and internal events. In ASD more importance is placed on adaptation rather than optimization.

Agile Modeling (AM) (Ambler, 2002) focuses on modeling practices and cultural principles. The models should be advanced enough to support both the design needs and purpose of the documentation. One aim of AM is to limit the amount of models and documentation. Clear communication and the organization of team structure is emphasized to address the cultural issues that may arise throughout the development process.

The Crystal Family (Cockburn, 1998) provides various methods which may be selected based on the particular project. It also includes principles by which the methods may be tailored to accommodate varying circumstances. Within the Crystal Family, a color coding scheme indicates the heaviness of the method. Therefore, the appropriate-colored method can be selected based on the size and critical nature of the project. This combination of methodologies is flexible enough to integrate the practices of other agile methods such as XP and Scrum.

Dynamic Systems Development Method (DSDM) (Stapleton, 1997) was first released in 1994. It is based on the concepts of rapid application development. The overall idea is to establish the amount of time and resources that are available and then determine the amount of system functionality based on these parameters. This is in contrast to the traditional approach of adjusting time and resources until the necessary functionality is reached.

Extreme Programming (XP) (Beck, 2005) is a lightweight process which attempts to address the constraints in software development and focuses on the ability to adapt to rapidly changing requirements. XP revolves around twelve key principles which include: system metaphor, planning game, small releases, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, forty-hour week, on-site customer, and coding standards. XP has become one of the most widely used agile methods and the subject of an increasing amount of research (e.g., Abrahamson, 2003; Abrahamson & Koskel, 2004; Alshayeb, 2005; Drobka, Noftz & Raghu, 2004; Fruhling & De Vreede, 2006; Lowell & Jeffries, 2004).

Feature-Driven Development (FDD) (Palmer & Felsing, 2002) emphasizes the design and building phases. It is very much iterative in nature, and focuses on generating working results every two weeks. Through the frequent delivery of tangible results risk is reduced and better quality can be ensured.

Internet-Speed Development (ISD) (Cusumano & Yoffie, 1999; Baskerville, Ramesh, Levine, Pries-Heje, & Slaughter, 2003) as the name implies is focused on the rapid release of software using short development cycles. ISD is premised on a management-oriented framework and is made up of time drivers, quality dependencies, and process adjustments. As Abrahamsson et al. (2003) noted, the “development is negotiated, compromised, and capricious as opposed to predefined, planned, and mutually agreed” (p. 3).

Pragmatic Programming (PP) (Hunt & Thomas, 2000) is built upon the concept of programming best practices. PP consists of 70 tips that focus on day-to-day problems. The practices are pragmatic in the sense that focus is placed on incremental and iterative development, with rigorous testing, and user-centered design.

Scrum (Schwaber & Beedle, 2002) is a team-based method for the management of development in volatile environments. It emphasizes flexibility, adaptability, and productivity from an empirical vantage point. Scrum adopts the iterative, incremental approach in order to respond to rapidly changing requirements. As opposed to other methodologies, it is scalable and leaves some flexibility in decisions regarding the implementation process.

Although numerous methods have been identified, two concepts lie at the heart of each of these methods: working code and effective people. Only through creativity, team work, customer participation, and continuous feedback can effective projects be completed on time and within budget (Highsmith & Cockburn, 2001).

Benefits and Challenges

As is typically the case with emerging development methodologies, there are proponents, opponents, and those who fall somewhere in between. Proponents of agile methods believe that traditional approaches are simply unable to respond quickly enough to changing system requirements (Erickson et al., 2005). Nerur, Mahapatra, and Mangalaraj (2005) echoed this sentiment by suggesting that current organizations are in need of “information systems that constantly evolve to meet their changing requirements – but the traditional, plan-driven software development methodologies lack the flexibility to dynamically adjust the development process” (p. 73). Although, Hilkka, Tuure, and Matti (2005) hypothesized that agile methods represented “old wine in new bottles” (p. 41), they concluded that the emphasis on user participation and the fact that these methods were well suited to younger organizational culture were positives.

Others have taken a more balanced approach to the use of agile methods by suggesting that in some cases combining aspects of traditional and agile methods may provide the best solution (Boehm, 2002; Lycett et al., 2003). For instance, Boehm stated that both traditional and agile methods have strengths and weaknesses and “outside of each approach’s home ground, a combined approach is feasible and preferable” (p. 64). Some simply pointed to the fact that the empirical evidence supporting the effectiveness of agile development methods is currently limited or lacking altogether (Abrahamson et al., 2003; Lindvall, Muthig, Wallin, Stupperich, Kiefer, May, & Kahkonen, 2004). Turk, France, & Rumpe (2005) investigated the underlying assumptions of agile methods and found that in multiple cases these assumptions were not upheld bringing into question the applicableness of agile methods in certain situations.

Because many believe that agile methods represent a new way of thinking, both benefits and challenges to the adoption of agile methods arise from a review of the literature. These benefits and challenges relate to the areas of requirements determination, project management, and implementation. Each of these areas will be covered in the following sections.

Requirements Determination

Sillitti, Ceschi, Russo, and Succi (2005) identified the handling of requirements as a primary difference between traditional and agile methods. The overall philosophy behind traditional methods has been that “efforts made in up-front planning activities and in artifact production will result in lower cost, timely product delivery, and better software quality” (Germain & Robillard, 2005, p. 17). Both Sillitti et al. (2005) and Boehm (2002) supported this statement as they characterized this “up-front” approach as descriptive of traditional methods. The rationale is that by gathering all requirements at the start of the project minimal changes are necessary once the actual development process begins. Agile methods, however, assume that change is inevitable and actually see change as an opportunity to improve the product. Therefore, agile methods worry only about the requirements necessary to meet immediate needs (Sillitti et al., 2005).

The study conducted by Sillitti et al. (2005) included sixteen companies in Italy. Eight of the companies used agile methods, the other eight used documentation-driven methods. Two basic questions were examined: (1) do agile and document-driven companies manage requirements variability in different ways, and (2) do agile and document-driven companies approach requirements gathering in different ways? (p. 3). The authors utilized a semi-structured survey questionnaire completed during an interview to gather the data. The results indicated that there are significant differences in the way document-driven and agile companies handle uncertainty in the areas of changing requirements, requirements problems, reasons for variation in requirements, how requirements are gathered, when and who gathers the requirements, and techniques for gathering. Overall agile companies appeared to be more flexible and customer-centric in regard to requirements. In sum, Boehm (2002) concluded, whereas traditional methods focus on thorough planning and detailed requirements determination, the primary objective of agile methods is the rapid development of working code.

Similarly, Highsmith and Cockburn (2001) argued that the primary goal of agile methods is to meet the needs of customers at delivery time rather than at the initiation of the project. Agile methods meet this goal by emphasizing the rapid development of working code over extensive time spent in project planning and model building. They suggested that due to the dynamic nature of current business environments traditional methods, which purport to anticipate all of the system requirements, were inadequate and respond too slowly. The key point from their perspective was that agile methods focus on features which the customer can understand, rather than tasks which are more in the realm of the developer. Through the development of working code and hiring of effective people, agile methods were geared to meet the needs of the changing business environment. Hunt and Thomas (2003) suggested that the challenge revolving around requirements determination calls for asking the question, “how do we decide to build something that isn’t strictly necessary today but could be cheaper in the long run to build (or start) now and could save us from larger pains later on” (p. 106)?

A final issue related to requirements determination in agile methods is quality. Huo, Verner, Zhu, and Babar (2004) investigated the question, “can agile methods ensure quality even though they develop software faster and can handle unstable requirements” (p. 520)? In order to answer this question they built outlines of the Waterfall model and agile methods and their associated quality assurance measures. They then proceeded to compare the two models to see if both employed similar measures of quality assurance. The results of the analysis of the models revealed three significant differences. First, some practices in agile methods serve both development and quality assurance purposes. Consequently, developers are involved in applying the quality assurance measures which is typically not the case in the waterfall method. Second, because agile methods emphasize continuous development, short iterations, and frequent feedback, the speed of two-way communications is faster than in the waterfall method. Third, agile methods have more dynamic techniques than the waterfall method. From these differences an argument could be made that agile methods do produce quality, in addition to speed and efficient handling of unstable requirements.

Project Management

In addition to requirements determination issues, agile methods also have a significant impact on project management methodologies. As Augustine et al. (2005) have suggested, traditional project management methods are now mismatched with emerging and dynamic systems. They identified control as the primary reason contributing to the difficulty of using traditional management methods with agile development teams. From their perspective, maintaining control is much simpler when the requirements of the system are well-defined, the problem is well-structured, and there is a heavy reliance on models and documentation.

In light of this statement by Augustine et al. (2005), Coram and Bohner (2005) argued that “agile methods offer a reasonable approach for the degree of change and uncertainty in today’s software development” (p. 370). Their perspective was that agile methods are well suited for projects that are ill-defined, new to the organization, or utilize leading-edge technology. Considering that implementation and project management are related, the authors identified several overlapping groups already mentioned by Boehm and Turner (2005) and Nerur et al. (2002) that are impacted in the management of agile projects. These included people, developers, testers, project leaders, customers, executive management, and the teams themselves.

Along the same lines as Boehm and Turner (2005) and Nerur et al. (2002), Coram and Bohner (2005) also suggested that developmental processes are also impacted. However, Coram and Bohner conceded that agile methods are not necessarily appropriate for every project, it may be in fact better to utilize traditional methods in certain situations. Boehm (2002) suggested that agile methods were most effective for smaller, dynamic projects instead of larger projects which tend to be better handled with traditional methods.

Implementation

The introduction of a new development methodology is many times accompanied by difficulties in actual implementation in addition to requirements determination and project management concerns. Organizations are often resistant to change and the move away from traditional methodologies (Boehm & Turner, 2005; Nerur et al., 2002). Consequently, Nerur et al. (2002) identified four key issues which must be addressed when adopting agile methods including: management and organizational, people-related, process-related, and technological. In regard to management and organizational issues they identified the biggest challenge as that of persuading the "project manager to relinquish the authority he/she previously enjoyed" (p. 76). For people-related issues the authors cited value and trust as important elements due to the team-oriented nature of agile methods. Process-related issues included the shift from the life-cycle model to a feature-driven model and finally, technological issues involved the development of tools to support agile development. Nerur et al. pointed out, however, that "tools alone cannot make software development successful" (p. 77).

Similarly, Boehm and Turner (2005) identified three critical challenges to implementation which coincide with Nerur et al. (2002). The challenges included development process conflicts, business process conflicts, and people conflicts. Development process conflicts involved variability, differing life cycles, legacy systems, and requirements determination. Business process conflicts included areas of human resources, progress measurement, and process standard ratings. Finally, people conflicts addressed management attitudes, logistical issues, pilot projects, and change management. A significant area of overlap between Nerur et al. (2002) and Boehm and Turner (2005) was in the area of people and processes. Cockburn and Highsmith (2001) pointed out that a core value of agile methods is the philosophy that people trump processes.

Further challenges were suggested by Cohn and Ford (2003) such as fear of micromanagement, concern about overzealous agile development teams, strained relationships between developers and testers, uneducated upper management, and uninformed human resource personnel. They summed up the issue by stating, “how an agile process is introduced into an organization will significantly impact the ultimate success of the process change” (p.78).

Global Software Development

With the rise in the globalization of business and the advancement of information and communication technologies, organizations are increasingly adopting global software development (GSD) as a strategy to meet the traditional budgetary and time constraints of software projects. According to Damian and Moitra (2006), GSD is “becoming the norm in the software industry” (p. 17). Carmel (1999) defined global software development as teams working together to accomplish project goals from different geographic locations. Outsourcing and distributed teams within the same organization that are located in different countries both represent examples of GSD (Layman, Williams, Damian, & Bures, 2006). The driving factors contributing to the movement toward GSD include: a large, talented global resource pool, proximity to the market, quick formation of virtual teams, “round the clock” development, cost advantages, and the need for flexibility (Carmel & Agarwal, 2001; Damian & Moitra, 2006; Herbsleb & Moitra, 2001). The result is that, “software development is increasingly a multisite, multicultural, globally distributed undertaking” (Herbsleb & Moitra, 2001, p. 17). With respect to the future of GSD, Carmel and Agarwal (2002) suggested that “software development projects will increasingly look like a global virtual archipelago with several clusters of colocated professionals sprinkled with dispersed individuals working remotely” (p. 29).

Benefits of Global Software Development

To some, GSD is becoming both a “business necessity” and a “pervasive business phenomenon” (Damian & Moitra, 2006, p. 18). However, not everyone is convinced that the formation of globally distributed teams is the answer to the ongoing challenges presented in software development. Ebert and De Neve (2001) advocated just the opposite, the building of “coherent and collocated teams of fully allocated engineers” (p. 63). Despite the presence of some opposition, the potential benefits of the 24 hour day, high-speed development, employment of more skillful staff, lower development costs, and the ability to quickly respond to the needs of local customers is a powerful lure to many organizations attempting to remain competitive in the software industry (Herbsleb & Moitra, 2001; Komi-Sirvio & Tihinen, 2005). The reality is that in many organizations a “geographically dispersed team might be the only way to gather all talented IT professionals around the world” (Carmel & Agarwal, 2001, p. 28).

Challenges of Global Software Development

In spite of the potential benefits of GSD, as development work becomes increasingly more virtual and distributed the challenges of working effectively in this environment will continue to increase as well. According to Komi-Sirvio and Tihinen (2005), distributed software development projects have many of the same challenges as colocated projects such as quality, schedule, and cost related problems. The only difference is that the distribution often exacerbates these problems. In a study conducted by Komi-Sirvio and Tihinen (2005), participants were provided with a list of eight different problem areas identified from prior research and ask to indicate which problems they had experienced. According to this study, the problems most encountered were within the development tools and environment area (81%). Under this area two main technical problems were predominant: network connectivity and compatibility of tools. Communication and contacts ranked second (74%), with cultural differences and physical distances repeatedly occurring. Design knowledge, ranked third (67%). The remaining areas are listed in ranked order: project management (59%), cultural differences (52%), time slippage/budget growth (52%), product management (33%), and communication tools (30%). The overall findings indicated a general consensus with the challenges identified by other researchers. In sum, the challenges of GSD can be broken down into the following categories: strategic, cultural, communication, geographic, knowledge management, project and process management, and technical (Battin, Crocker, Kreidler, & Subramanian, 2001; Herbsleb & Moitra, 2001; Herbsleb & Mockus, 2003; Komi-Sirvio & Tihinen, 2005).

Strategic

Strategic issues can involve how to divide work across distributed sites as well as dealing with overall resistance to the implementation of GSD. Strategically, the ideal arrangement would, to a large degree, allow each site to work independently while still fostering flexible and effective communication (Herbsleb & Moitra, 2001). An example of a strategic decision in regard to the division of work is the formation of bridgehead teams, where 75 percent of the team is located offshore, while the other 25 percent resides onshore (Carmel & Agarwal, 2001). A solution proposed by Carmel and Agarwal (2001) in relation to distributed work recommends reducing intensive collaboration between the Center (a firm in North American, the European Union, or Japan) and the Foreign Entity. As the Foreign Entity takes on more and more ownership of a particular project, there is less interaction with the Center, thus the intensity of ongoing collaboration is lessened and the problems associated with distance are decreased.

Battin et al. (2001) echoed the idea of reducing intensive collaboration with the expression, “distribute entire things for the entire lifecycle” (p. 76). By assigning each team full lifecycle responsibility they are able to provide support for their own parts and communication problems caused by the division of work can be reduced. Although this strategy may work effectively for some organizations, for others, differences in resources, expertise, and infrastructure, can still pose significant hindrances to the success of GSD (Battin et al., 2001; Herbsleb & Moitra, 2001).

In addition, if the organization as a whole does not fully buy into the concept of GSD, success is made even more difficult. Resistance may come in the form of differences between middle and upper management on the reasons for adopting GSD and the benefits it provides. Resistance may also come at the individual level. For example, the transition to GSD may instill a sense of fear in current employees who worry that their jobs will be eliminated when the organization distributes the work to another location (Herbsleb & Moitra, 2001).

Cultural

It is widely accepted that cultural differences present a significant challenge to GSD (e.g., Damian & Moitra, 2006; Herbsleb & Moitra, 2001; Carmel, 1999; Evaristo, Scudder, Desouza, & Sato, 2004). Culture can have a great impact on how individuals interpret and react to various situations (Kotlarsky & Oshri, 2005). Factors such as the need for structure, attitude toward organizational hierarchy, sense of time, language barriers, and overall attitude toward international development all come into play in some way (Herbsleb & Moitra, 2001; Battin et al., 2001). Culture is complex in that it may involve both organizational and national dimensions (Carmel & Agarwal, 2001; Holmstrom et al., 2006). As defined by Carmel and Agarwal (2001), organizational culture "encompasses the unit's norms and values, where the unit could range from a small technology company to a multinational enterprise" (p. 25). A dimension of organizational culture would be the systems development culture, with its use of specific methodologies and project management practices. National culture, on the other hand, was defined as "encompasses an ethnic group's norms, values, and spoken language, often delineated by political boundaries of the nation-state" (p. 25). In order for GSD to be successful, both of these dimensions of culture must be taken into consideration.

Carmel and Agarwal (2001) strongly advocated the use of a cultural liaison that travels back and forth between the Center (onshore) and the Foreign Entity (offshore) to alleviate the problems associated with organizational culture. The role of the liaison is to "facilitate the cultural, linguistic, and organizational flow of communication and to bridge cultures, mediate conflicts, and resolve cultural miscommunications" (p. 27). In a study conducted by Carmel (1999), 47% of global software development teams had a person who had the characteristics of a cultural liaison. Battin et al. (2001) concurred with the importance of the liaison by identifying that person as "a critical success factor for global development" (p. 76). For alleviating the differences in national culture, spoken language was identified as a crucial factor, and was noted as "one of the reasons for the success of offshore IT work in countries with strong English language capabilities such as the Philippines and Singapore" (Carmel & Agarwal, 2001, p. 27). In a case study conducted by Holmstrom et al. (2006), it was also found that cultural issues impede communication, which increases the potential for misunderstandings which ultimately may lead to the failure of GSD projects (Herbsleb & Moitra, 2001).

Communication

The software development process is complex and is highly dependent on effective communication, even among colocated teams. The importance of communication is heightened even further among geographically distributed teams. The loss of "communication richness" is a significant problem often caused by the physical distance and time zone differences engendered by GSD (Battin et al., 2001). As such, it is extremely important that protocols be established for facilitating both official and informal communication. Official communication may be distributed via the organization's intranet, email, and in some cases, telephone calls. Informal communications, which often takes place around the coffee pot or across a cubicle is much more difficult to imitate in a GSD environment. This lack of informal communication may lead to issues that "go unrecognized or lie dormant and unresolved for extended periods of time" (Herbsleb & Moitra, 2001, p. 18). Fortunately, as advances in information and communication technologies such as instant messaging, audio and video conferencing, and groupware applications continue, the difficulties encountered in informal communication may be alleviated.

Geographic

The combination of communication and geographic distance represent a significant challenge to GSD. Two of the critical areas in which distance creates difficulty are coordination and control. Coordination may be defined as the integration of "each task with each organizational unit, so the unit contributes to the overall objective"; whereas, "control is the process of adhering to goals, policies, standards, or quality levels"; and "communication is a mediating factor affecting both coordination and control" (Carmel & Agarwal, 2001, p. 23). Thus, communication serves as a crucial intermediary between coordination and control in globally distributed environments. According to Carmel and Agarwal (2001), therefore, the main challenge in global software development was summarized as follows: "distance negatively affects communication, which in turn reduces coordination effectiveness" (p. 23). Similarly, Herbsleb and Moitra (2001) argued that because multi-site development typically takes significantly longer than comparable colocated tasks, the interplay between communication, coordination, and control is extremely important in the amount of delay caused by geographic distance. With respect to the issues of distance and communication, Herbsleb and Mockus (2003), stated that, “in contrast to the frequent interaction of colocated work, there is very convincing evidence that the frequency of communication generally drops off sharply with physical separation among coworkers’ offices and that the sphere of frequent communication is surprisingly small” (p. 481).

In a study conducted by Herbsleb and Mockus (2003) it was found that work distributed across sites appears to take two and one-half times longer than similar projects where the entirety of the work is done in a colocated environment. The study showed that that size, diffusion, and number of people were all directly related to the delay. Interestingly, however, there was no direct link between the amount of delay and the distributed nature of the work. This suggested, therefore, that due to the greater number of people involved in distributed projects as opposed to colocated projects, the number of people working on the project had a more significant influence on the amount of delay than the geographic distance. This finding may be important when determining the structure of teams working in globally distributed environments. There is evidence that suggested that as a team grows in size, team productivity actually decreases rather than increases (Hackman, 2002).

Another facet of geographic distance relates to temporal distance or time zone difference. The reduction of temporal distance represents a ”trade-off between the advantages and disadvantages of synchronous and asynchronous communication” (Carmel & Agarwal, 2001, p. 27-28). Asynchronous communication typically includes email, voice mail, discussion groups, and groupware. Synchronous communication consists of telephone calls, audio and video conferencing, application sharing, and instant messaging. Although asynchronous communication technologies have become a part of normal life, Carmel and Agarwal (2001) argued that there are numerous advantages for incorporating synchronous communication into the GSD process. These include the resolution of miscommunications and misunderstandings, as well as taking care of small problems before they become big problems. Increased delays and making problems more complicated are major problems with asynchronous communication. Carmel and Agarwal quickly noted, however, that reducing temporal distance via synchronous communication is “no panacea”, because it too has its own limitations.

Also related to geographic distance is the problem encountered when vendor support is not available at all locations. Vendor support takes into consideration the version each site is using as well as whether or not there is local technical support. An additional issue to consider are the governmental regulations dealing with work laws, visas when traveling, import/export rules, and customs regulations (Battin et al., 2001). Finally, integrating globally distributed teams into a coherent team and instilling a sense of “teamness” is a challenging endeavor (Battin et al., 2001; Carmel, 1999; Herbsleb & Mockus, 2003).

Knowledge Management

As noted by Kotlarsky and Oshri (2005), however, technical approaches must be coupled with social aspects such as building rapport and trust and knowledge sharing in order for globally distributed system development projects to be successful. Inadequate knowledge-sharing mechanisms and poor documentation may impede an organization from taking advantage of the full benefits of GSD (Herbsleb & Moitra, 2001). As recognized by Desouza, Awazu, and Baloh (2006), in globally distributed environments “knowledge moves within and across organizational boundaries, and it must do so in an effective and cost-efficient manner for the superior calibration of software products and services” (p. 30).

Project and Process Management

Project and process management issues involve lack of synchronization and risk management. One of the benefits of GSD is the 24-hour day. This takes place as one team “hands off” the work that they have accomplished during their day to another team which is just starting their day. If methods are not put into place to synchronize this hand-off the result may lead to redundant work, misunderstanding, and an overall delay in project completion. In regard to risk management many organizations fail to consider the cultural differences between distributed sites and implement traditional techniques which may or may not be appropriate in a global setting (Herbsleb & Moitra, 2001).

Technical

Technical issues including slow and unreliable networks, configuration management, incompatible data formats, system architecture, software integration, and different tool versions may lead to coordination breakdowns and thus a delay in the development process (Battin et al., 2001; Herbsleb & Moitra, 2001). Komi-Sirvio and

Tihinen (2005) found in their study that the challenges most encountered were technical in nature and included problems with network connectivity and compatibility of tools.

Virtual Teams

Townsend, DeMarie, and Hendrickson (1998) wrote of the development of a new workplace that would be “unrestrained by geography, time, and organizational boundaries”; it would be “a virtual workplace, where productivity, flexibility, and collaboration will reach unprecedented new levels” (p. 17). This new virtual workplace would be facilitated by the formation of virtual teams, those “groups of geographically and/or organizationally dispersed coworkers that are assembled using a combination of telecommunications and information technologies to accomplish an organizational task” (p. 18). Similarly, Lipnack and Stamps (1997) defined a virtual team as “a group of people who interact through interdependent tasks guided by common purpose” and who work “across space, time, and organizational boundaries with links strengthened by webs of communication technologies” (p. 6).

Townsend et al. (1998) identified five factors for the move to virtual teams: 1) the increasing prevalence of flat or horizontal organizational structures; 2) the emergence of environments that require interorganizational cooperation as well as competition; changes in workers’ expectations of organizational participation; 4) a continued shift from production to service/knowledge work environments; the increasing globalization of trade and corporate activity (p. 18). In general, virtual teams are characterized by the following features: reliance on information and communication technologies, flexible composition, temporal nature, the ability of members to cross over organizational, professional, and cultural boundaries as well as time constraints (e.g., Jarvenpaa & Leidner, 1999; Lipnack & Stamps, 1997; Maznevski & Chudoba, 2000).

In the virtual team literature the following types of teams were identified: colocated, virtual, and global. In an attempt to clarify the difference between these types of teams McDonough, Kahn, and Barczak (2001) suggested the following definitions: colocated teams are “comprised of individuals who work together in the same physical location and are culturally similar”; virtual teams are “comprised of individuals who have a moderate level of physical proximity and are culturally similar”; global teams are “comprised of individuals who work and live in different countries and are culturally diverse” (p. 111).

Global Virtual Teams

Based upon the characteristics of working and living in different countries and the diverse cultural makeup of the team, Powell, Piccoli, and Ives (2004) stated that global virtual teams represent a particular type of virtual team. Maznevski and Chudoba (2000) defined global virtual teams as “internationally distributed groups of people with an organizational mandate to make or implement decisions with international components and implications” (p. 473). The emphasis in this definition is again on the “international” emphasis of global virtual teams. An another definition of a global virtual team stated that it is “an example of a new organization form, where a *temporary team* is assembled on an as-needed basis for the duration of a task, staffed by members from far corners of the world” (Jarvenpaa, Knoll, & Leidner, 1998, p. 30).

The following characteristics of global virtual teams, therefore, have been identified: 1) they are identified by their organization and members as a team; 2) they are responsible for making and/or implementing decisions important to the organization's global strategy; 3) they use technology-supported communication substantially more than face-to-face communication; 4) they work and live in different countries (Maznevski & Chudoba, 2000 p. 473). In general, global virtual teams like virtual teams are characterized by the ability to traverse the boundaries of time and space (Lipnack & Stamps, 2000), however, global virtual teams are also distributed internationally and culturally diverse (Jarvenpaa & Leidner, 1999; Maznevski & Chudoba, 2001).

Challenges of Virtual Teams

There has been a considerable amount of research carried out in the area of virtual teams and several studies specifically investigating global virtual teams (e.g., Jarvenpaa & Leidner, 1999; Kayworth & Leidner, 2000, 2001-2002; Maznevski & Chudoba, 2001). Because of the distributed nature of these virtual teams, they share many of the challenges associated with global software development including: logistical, cultural, technical, and communication (Dube & Pare, 2001; Kayworth & Leidner, 2001-2002).

Logistical

Space, place, and time are considered important logistical elements in the use of virtual teams (Sarker & Sahay, 2004). Logistics address the difficulties related to the scheduling of meetings and the arrangement of travel when team members are distributed across geographic distance and differing time zones (Kayworth & Leidner, 2000; Sarker & Sahay, 2004; Solomon, 1995). As with global software development, virtual teams offer the possible advantage of a 24-hour workday, but need to put policies in place for the "handing off" of work from one team member to another in such a way to reduce time delays, redundancy of tasks, and misunderstandings (Herbsleb & Moitra, 2001).

Cultural

The impact of culture on virtual teams can be significant. This impact may be felt in the way individuals "perceive information, act upon it, and relate to other individuals (Kayworth & Leidner, 2001, p. 187). Research suggested that differences in culture may lead to challenges in both coordination and communication between virtual teams (e.g., Johansson, Dittrich, & Juustila, 1999; Kayworth & Leidner, 2000; Maznevski & Chudoba, 2000; Sarker & Sahay, 2004; Van Ryssen & Godar, 2000).

A study of three virtual teams comprised of graduate and undergraduate students located in the United States, Mexico, and Europe indicated that cultural differences had a significant affect on communication and coordination. The language barrier, especially between the American and Mexican students was identified as the most common cultural issue. A second cultural issue which surfaced was the concept of urgency of timing to complete the project. Through an analysis of the American student's comments, it was found that they considered some cultures to be more relaxed than others and felt it was necessary that the other teams be reminded of these differences when it came to completing the project in a timely manner (Kayworth & Leidner, 2001).

In another study conducted among student virtual teams distributed in Sweden and Finland, it was found that both of these particular cultures displayed a tendency to avoid conflict in order to keep the other members happy. Consequently, communication was hampered because neither side wanted to admit that there was a problem. Due to the individualistic nature of Swedish and Finnish cultures, there was a sense of fear that communicating truthfully about difficulties in the project or the fact that things were not going well, would lead to the lose of self-respect or feelings of guilt (Johannson et al., 1999).

A study of three cross-functional virtual teams in one large company found that communication was also hindered by something as seemingly simple as the difference of accents between team members located in the southern and northern regions of the same company. This subtle cultural difference appeared to be significant enough to cause communication obstacles (Robey, Khoo, & Powers, 2000).

Sarker and Sahay (2004) in a study of eight virtual software development teams comprised of students from an American and a Norwegian university found multiple difficulties associated with cultural differences. These included the choice of the preferred language and the associated competency issues; misunderstandings caused by different conversational styles; discomfort when Norwegian members switched back and forth between English and their native language; and the lack of consideration of national holidays when establishing performance expectations and project goals.

Interestingly McDonough et al. (2001) found that several areas which would be thought to be affected by cultural differences such as goal setting, budgeting, aligning schedules, managing resources, and identifying needs were actually related to the geographic distance between members more so than the differences in culture.

Communication

As with any team, virtual or otherwise, communication is a crucial element. Two studies in particular found that virtual teams actually communicated more often than colocated teams (Eveland & Bikson, 1988; Galegher & Kraut, 2004). However, due to a heavy reliance on information and communication technologies rather than face-to-face interactions, virtual teams face a potential greater challenge than colocated teams when attempting to interact, share and interpret meaning, and come to agreement (Kayworth & Leidner, 2000). As discussed previously, logistical, technical, and cultural considerations all play a role in the development of communication channels in geographically distributed work. The lack of mutual knowledge and understanding and the lack of a common language have both been cited as creating hindrances to communication efforts (Crampton, 2001; Qureshi & Vogel, 2001). Kayworth & Leidner (2001-200) cited ineffective leadership as another common reason for lack of effective communication among members.

Issues such as relationship building, coordination, trust, and cohesion are all predicated on effective communication between team members (e.g., Jarvenpaa et al., 1998; Jarvenpaa & Leidner, 1999; Lipnack & Stamps, 2000; Lurey & Raisinghani, 2001). Consequently, communication should not simply be limited to the scope of the project itself. Social communication should also be encouraged within the team. Studies have shown that this social aspect of communication leads to higher levels of trust and stronger social and emotional relationships (Jarvenpaa & Leidner, 1999; Robey et al., 2000).

It should be noted, however, that although these challenges are faced by both virtual and global virtual teams, some have argued that global teams may face increased difficulties due to greater geographic dispersion and greater cultural and language diversity (McDonough et al., 2001). Dube and Pare (2001) summarized this sentiment by stating that global virtual teams “confront significant challenges over and above more localized virtual teams” (p. 71).

Technical

As with global software development, advances in information and communication technologies have greatly enabled the formation of virtual teams. However, many technical difficulties may be encountered such as the inability to send and receive email from one site to another, network limitations, servers going down, the time delay involved in both asynchronous and synchronous communication tools, and the inherent overall limitations of technology (Kayworth & Leidner, 2000). In addition to the challenges involved with the technology itself, research suggested that variation in the technical skills of team members as well as an inability to deal with various technical problems may negatively impact the satisfaction level of individual members as well as the overall team experience and performance (Dube & Pare, 2001; Kayworth & Leidner, 2000; Van Ryssen & Godar, 2000). As noted by Townsend et al. (1998), the “real challenge of virtual team effectiveness is learning how to work with these new technologies" (p. 22).

One study in particular indicated that the newness of the technology being employed had more affect on the team members than the change to the virtual team structure (Hollingshead, McGrath, & O'Connor, 1993). However, in relation to team structure and the heavy reliance on technology to facilitate communication and coordination, a significant difference between traditional teams and virtual teams is the "amount of technical training that is required to empower the team member to function in the virtual environment" (Townsend et al., 1998, p. 27). Furthermore, Sarker et al. (2001) found that within the virtual team structure, each member's experience with technology impacted the process of developing, internalizing, and adapting to external norms.

Virtual Teams and Global Software Development

Although global software development (GSD) and virtual teams constitute two distinct areas of research, the two are becoming more and more intertwined. As evidenced by the review of literature both share many of the same benefits and challenges. With the move toward GSD by many organizations, the pairing of virtual teams and globally distributed projects is becoming more common (Sarker & Sahay, 2004). In today's turbulent business environment, virtual teams enable greater organizational flexibility and the ability to respond quickly to change. This is particularly important in global software development because of rapidly changing system requirements. Research investigating the use of virtual teams in the area of software development has been conducted and several representative studies are summarized below.

Use of Videoconferencing as a Support Mechanism

Andres (2002) assessed the usefulness of videoconferencing as a support mechanism for geographically dispersed software development teams. The study sought to compare a virtual team supported by videoconferencing with an equivalent face-to-face team in the areas of team productivity, perceived interaction quality, and group process satisfaction. It was hypothesized that the face-to-face team would experience greater effect in all three areas. The results showed that in the face-to-face environment team productivity and perceived interaction quality were statistically significant. This result indicated that teams using face-to-face communication experienced greater team productivity and perceived interaction quality than virtual teams utilizing videoconferencing technology. However, the results for group process satisfaction were not statistically significant different between face-to-face and virtual teams.

Examination of Communication, Coordination, and Satisfaction

Barkhi, Amiri, and James (2006) examined the communication, coordination, and satisfaction of virtual software development team members in both colocated and remote settings. Based on previous research the authors proposed that participants in software development teams: 1) are more likely to collaborate with colocated rather than with remote members and 2) to break communication with remote than with colocated members; 3) perceive more difficulty in coordination of tasks with remote rather than with colocated members; 4) are more likely to shift blame to remote rather than colocated members; and 5) can gain satisfaction with the distributed work process if the can work effectively with remote members.

The study was conducted over a 14-week period utilizing two groups of undergraduate students. One group was located in the Eastern part and the second in the Southern Midwestern part of the United States. The project involved the development of a database application and the accompanying documentation. The team structure was comprised of four team members, two at each location. Face-to-face communication could take place between the colocated members, but all communication with remote team members was required to take place via a web-based information and communication technology tool. Overall there were 82 participants, 44 from one site and 38 from the other. After the initial formation of teams, the students could break the formal guidelines of the four member team if they could provide justification for it. The resulting teams included one student from each site working alone, 29 and 23 students respectively working face-to-face only, zero participants working remote only, and 14 participants from each site working face-to-face and remotely. The study employed both quantitative and qualitative methods. Quantitative methods were used to calculate the ratio of team structures to total participants. A qualitative method using open-ended questions was used to gather data regarding the team member’s experiences in the project (Barkhi et al., 2006).

Through an interpretive analysis of the data gathered via open-ended questions, the results for each proposition indicated the following. Due to delays, slow feedback, and a general lack of personal relationships being developed, the data supported both proposition 1 and proposition 2, that team members would tend to collaborate more with colocated members than remote members and would be more likely to break communication with the remote members. Team comments also indicated support for proposition 3 and proposition 4, which dealt with participants perceptions of the coordination difficulties and the tendency of colocated members to blame problems on the remote members. Finally, it was found that those distributed teams that did learn to work together considered their experience satisfying. The authors concluded that the use of lean information and communication technologies can impact the success of distributed software development teams. However, by reducing ambiguities, developing relationships, and defining team roles, virtual software development teams can be successful and satisfying. In sum, the authors stated that ”a virtual software development team is a fragile structure and should be managed with care” (Barkhi, et al., 2006, p. 58).

Relationships between Geographic Dispersion, Processes, and Effectiveness

Crampton and Webber (2005) investigated the relationships among geographic dispersion, team processes, and effectiveness in software development teams. The study was conducted among 218 members of 39 software development work teams at an international consulting firm which specialized in the development of custom software and systems integration. Specifically, the authors addressed the following hypotheses in regard to geographic dispersion: 1) it will have a negative affect on work processes so that dispersed team members will have less effective processes than colocated teams; 2) it will negatively effect team performance so that dispersed teams will show less effective performance than colocated teams; and 3) the relationship between team geographic dispersion and team performance will be mediated by team work processes. The research method included qualitative and quantitative approaches. Initial semi-structured interviews were carried out among 10 project managers and company executives. The purpose of these interviews was to gather information to assist in the development of a formal survey. 500 surveys were distributed electronically, 246 were returned, and 218 were eventually deemed acceptable for use representing 39 teams. The unit of analysis was the work team itself. Regression analysis was utilized to test the direct relationships between geographic dispersion, team processes, and perceived team performance.

The results indicated support for both hypotheses 1 and 2 that dispersed members have less effective work processes than colocated members and that geographic dispersion has a statistically significant affect on perceived performance. The results indicated some support for hypothesis 3, which revealed that team processes partially mediated the relationship between geographic distance and perceived performance. Based upon the findings, the authors concluded that “teams with geographically dispersed members face significant challenges in developing and maintaining a social system that connects individuals to their tasks and each other” (Crampton & Webber, 2005, p. 763).

Effectiveness in Software Engineering Projects

Edwards and Sridhar (2003) conducted an analysis of the effectiveness of global virtual teams in software engineering projects. The study was conducted among student teams located at the University of Western Ontario, Canada and the Indian Institute of Management, Lucknow, India. In all 24 distinct teams participated in the study which took place over a five week period of time. The purpose of the study was to examine which factors affected the quality of requirements definition artifacts and to determine the effectiveness of global virtual teams to perform this type of task.

In order to facilitate this study, Edwards and Sridhar (2003) developed a theoretical framework consisting of the following constructs. The framework consisted of seven predictor variables and four outcome variables. The predictor variables included: ease of use of technology, structure of project tasks, effects of time difference, trust between teams, difference in academic orientation of teams, difference in cultural orientation of teams, and size of the teams. The outcome variables included learning effectiveness, quality of projects, virtual team project experience, and effect on software engineering process. A 31 item survey instrument was developed and validated for reliability with all constructs having a Cronbach’s alpha closer to or greater than .70.

In order to test the relationship between the predictor variables and the outcome variables Pearson’s product-moment correlations were performed. Interestingly, the results indicated that time, culture, and team size did not show statistical significance with any of the outcome predictors. In addition the quality of the project also showed no statistically significant relationship with any of the predictor variables. Also of importance was the result that trust was a statistically significant predictor of all four outcome variables. Ease of use of technology and structure of project team tasks were statistically significant predictors of three of the four outcome variables including learning effectiveness, virtual team project experience, and effect on software engineering process. Academic orientation was a statistically significant predictor of virtual team project experience and effect on software engineering process. The results of the correlations between the outcome variables indicated statistically significant relationships between learning effectiveness and virtual project experience and the effect on software engineering process.

Finally, it was revealed that there was a statistically significant relationship between project experience and effect on software engineering process. The overall results indicated that ease of use of technology, trust, and well-defined tasks had a positive influence on the efficiency, effectiveness, and satisfaction level of the global virtual teams (Edwards & Sridhar, 2003).

Agile Methods and Globally Distributed Environments

Currently, there is limited research exploring the use of agile methods in globally distributed environments. However, as organizations consider the move to global software development and the use of virtual teams in order to facilitate flexibility and adaptability, there appears to be a growing interest in the issue of whether or not distributed software development as a whole can be agile (Agerfalk & Fitzgerald, 2006; Ramesh et al., 2006) and whether or not specific practices such as pairing programming can be effective among globally distributed teams (Flor, 2006). This issue revolves around the significant difference in key tenets of agile and distributed development approaches. Agile methods tend to follow more informal processes, while distributed development relies heavily on formal mechanisms (Ramesh et al., 2006). Consequently, up to this point, the prevailing viewpoint has been that “agile methods are not applicable for GSD” (Holmstrom et al., 2006, p. 8).

A further reason contributing to this view is that proponents of agile methods insist that agile practices must be used in their entirety in order to be effective (Fitzgerald et al., 2006). One of the overriding practices is the use of colocated teams and the principle that the “most efficient and effective method of conveying information to and within a development team is face-to-face conversation” (Fowler & Highsmith, 2001). Obviously, this poses an immediate problem to the application of agile methods in globally distributed environments.

Because of the emerging nature of the study of agile methods in distributed environments, the research that is available is a combination of empirically-based studies as well as experience reports. The following section briefly summarizes the current research as it relates to the use of agile methods in globally distributed environments.

Achieving Good Communication

Xiaohu, Bin, Zhijun, and Maddineni (2004) examined the question of how to achieve good communication for global software development (GSD) when using Extreme Programming (XP). The study consisted of a software development team located in China and a customer located in the United States. The authors identified multiple types of communication streams that occur in software development: 1) between the team member and the customers; 2) between the team members and the project manager; 3) between the individuals that make up the customer and; 4) between the members of the team.

As with related studies on the use of XP regardless of the environment, communication is crucial for success. In agreement with Holmstrom et al. (2006), the authors cited delays in response time as a significant challenge when attempting to apply agile methods in distributed environments. In this study, the use of the XP practice of small releases served to reduce the complexity of the project and to increase the rate at which the development team was able to respond to the customer's request. In addition the story cards used as a part of the planning game served to reduce delays by further facilitating communication between team members and the customer.

As noted by Schummer and Schummer (2001), the two key issues for utilizing the planning game in distributed environments is to ensure that the story cards are remotely accessible and that proper information and communication technologies are in place to allow communication between team members as well as the customer. In this specific instance, the development team was divided into two groups: the reverse engineering group which was responsible for providing the customer with system requirements and the forward engineering team which received the confirmed requirements from the customer and began work on the system. The entire process, therefore, was iterative in nature with requirements being sent back and forth between the reverse engineering group, the customer, and the forward engineering group. This series of small iterations served to alleviate the delay in response between the development teams and the customer. The combination of the reverse-engineering team and the customer attempted to replicate the XP practice of the on-site customer, and the forward engineering group played the role of the original programmer. Thus, “the iterative developments reduced the impact of the communication delay since development of iteration can be done in parallel with the confirmation of the requirement for the next iteration” (Xiaohu et al., 2004, p. 1848).

Overall, this approach served to create a communication shortcut. The authors acknowledged that XP practices are indeed difficult to implement in globally distributed environments because of delays in customer feedback. The conclusion of the study, however, suggested that although implementation was difficult, XP can be successfully used within GSD (Xiaohu et al., 2004).

Reducing the Influence of Distance on Communication, Coordination, and Control

One of the major challenges in global software development is distance. It is this distance that contributes to problems in the areas of communication, coordination, and control (Carmel & Agarwal, 2001; Herbsleb & Mockus, 2003). The problem of distance significantly impacts the application of agile methods in globally distributed environments. Holmstrom et al. (2006) conducted two in-depth case studies which examined the question, “can agile methods be used to reduce the negative influence of distance on communication, coordination, and control in a GSD context” (p. 11)? In order to examine this question, three specific types of distance were identified: temporal, a measure of the dislocation in time experienced by two actors wishing to interact; geographical, a measure of the effort required for one actor to visit another; and, sociocultural, a measure of an actor’s understanding of another actor’s values and normative practices (Agerfalk et al., 2005). The definitions for communication, control, and coordination were borrowed from Agarwal and Carmel (2001).

The cases consisted of Intel and Hewlett-Packard which are both headquartered in the United States with development teams in Ireland. The agile methods utilized were a combination of Extreme Programming (XP) (Beck, 2000) and Scrum (Schwaber & Beedle, 2002). Interviews revealed that time zone differences and delayed responses were the most significant challenges in relation to temporal distance, whereas, geographical distance inhibited the development teams from building a feeling of trust and “teamness”. Finally, sociocultural distance involved the barriers of language including interpretation and meaning, differences in culture, politics, and religion.

In reference to the issues of temporal, geographical, and sociocultural distance, the study indicated the following: the XP practice of pair programming helped to raise the level of individual responsibility among distributed team members to create overlapping times when they could work together, thus reducing the amount of temporal distance; Scrum planning practices helped to foster a sense of “teamness” among the distributed members with the result of alleviating the problems associated with geographical distance; lastly, practices such as pair programming, and Scrum’s pre-game phase improved the overall communication between distributed team members and helped to lessen the problems associated with sociocultural distance. The overall result was the decrease of the negative effect of distance on communication, coordination, and control. In sum, Holmstrom et al. (2006) concluded that “contrary to previous research, our findings suggest that agile methods may be more amenable to GSD than has been previously reported” (p. 16).

Necessity for Colocation in XP Teams

Kircher, Jain, Corsaro, and Levine (2001) examined the question of whether it is really necessary for team members using XP to be physically located next to each other. In response to this answer, Distributed Extreme Programming (DXP) was developed, which takes the basic principles of XP and applies them in a distributed team environment. The authors defined DXP as “eXtreme Programming with certain relaxations on the requirements of close physical proximity of the team members” (p. 2). Although the authors claimed that DXP addresses all of the XP practices in some way, they separated the practices that require colocation from those that do not and emphasized that it is these practices that must be addressed in order for DXP to be successful in distributed environments. According to the authors the following practices required colocated teams: planning game, pair programming, continuous integration, and on-site customer. Those practices not requiring colocated teams included: small releases, system metaphor, simple design, testing, refactoring, collective ownership, forty-hour week, and coding standards. Furthermore, for DXP to be effective the following tools and conditions are assumed to exist: connectivity, email, configuration management, application sharing, video conferencing, and familiarity.

In line with Holstrom et al. (2006), the areas of communication, coordination, and control were identified as significant challenges to the implementation of XP practices in distributed environments. In addition to these three, the authors also cited infrastructure, availability, and management issues. In order to test the effectiveness of DXP, an experiment was conducted which enlisted four team members in globally distributed locations (India, Germany, United States) to complete a software development project.

One team member played the role of customer, while the other three members represented the software development team. In order to facilitate the planning game, multiple videoconference sessions were conducted with the customer to determine requirements via story cards. Application sharing software was used to enable each member to participate in the session. For pair programming, the story cards were assigned to two distributed programmers and extensive use of videoconferencing and application sharing were utilized. A configuration management tool was used to enable continuous integration. Finally, videoconferencing and email were used to communicate with the distributed customer.

It was found that using a "combination of synchronous communication, such as videoconferencing and asynchronous communication, such as e-mail, to be the most effective" way to communicate (Kircher et al., 2001, p. 5). However, the experiment revealed that it was very difficult to simulate the physical presence or awareness afforded in a colocated environment (Schummer & Schummer, 2001). In conclusion, the authors suggested that modifying XP practices that require colocation, i.e., DXP, can be efficiently used within a distributed environment.

Colocation, Distribution, and XP Teams

Schummer and Schummer (2001) also addressed the issue of colocation, distribution, and the use of XP teams. As noted by the authors, XP requires rich communications and a high degree of awareness among the development team. These issues are crucial to the success of distributed XP teams. Three specific challenges were identified: 1) hindrance of communication within the team; 2) less awareness among the team members and; 3) common access to physical objects and places is difficult.

In an attempt to address these issues, Schummer and Schummer (2001) categorized the following XP practices as “communication centered”: the planning game, testing, pair programming, continuous integration, and communication with the external customer. Comparing this list to the one created by Kircher et al. (2001) for those practices which require colocation, only testing represents a difference. It is these communication centered practices that challenge the application of agile methods to globally distributed environments and that must be addressed.

To alleviate the problems caused by communication and awareness, Schummer and Schummer (2001) suggested that metrics be employed to gauge the progress of the team members and to communicate the state of the project. The authors also suggested that groupware applications be employed to raise the level of awareness among team members and enhance communication, coordination, and collaboration. Communication technologies such as audio conferencing systems could be employed among distributed pair programming teams and video conferencing systems could be used to connect the external customer with the team. Workflow management systems, document management systems, and version management systems were listed as options to increase the coordination efforts between the team. Collaboration could be enhanced by the utilization of group decision support systems, multi-user editors, application sharing programs, and desktop conferencing. In conclusion, Schummer and Schummer argued that “with the appropriate technical support at hand – XP might work as well in distributed settings” as in colocated environments (p. 2).

24x5 Around-the-Clock XP Programming Team

Yap (2005) presented an experience report of a global company spanning three geographically distributed regions: the United States, the United Kingdom, and Singapore. The goal was to form a 24x5 around-the-clock Extreme programming team. In line with other literature related to global software development and the use of virtual teams, the challenges of communication, cultural differences, geographic distance, and technical issues were encountered (e.g., Battin et al., 2001; Herbsleb & Moitra, 2001). In order to alleviate these typical problems, the company enlisted an extreme programming coach within each geographically distributed region. It was the responsibility of the coach to not only train team members, but also to assist local business units to adjust to the new practices. Consequently, the coach became an integral member of the team.

The company also engendered trust within the teams by organizing Extreme Programming boot camps to which developers were sent. Within this face-to-face environment, team members were able to work together, build an initial amount of trust, and then take that experience back to their respective global site. Finally, the company enlisted a “rotating guru”, a senior team member, to visit the other sites to assist with infrastructure issues and to provide additional training and mentoring. This appears to be similar to the idea of the cultural liaison (Carmel & Agarwal, 2001).

Even after implementing coaches, boot camps, and gurus, the company faced continuing challenges in the transition process. As Yap (2005) noted, “the global team struggled to share and maintain common process vision and values. To share the same code base and system, we had to trust each other as equal team members” (p. 4).

When introducing new technologies or architectural changes, the company found that the following approaches worked rather effectively: 1) daily handovers, which became a teaching mechanism for sharing lessons learned, collaboration of ideas, and addressing cross-regional issues; 2) the use of video conferencing tools to facilitate face-to-face communication during the daily handover process; 3) installation of the same tools and environments across pairing machines; 4) frequent remote pairing sessions; 5) establishing a mechanism for gaining team “buy-in” for new processes and technologies; 6) not becoming too rigid by allowing some flexibility in each region based on culture.

Overall, it was found that “it is very important to have a ‘balanced’ team, that is, each region should be equal in size and technical skill level” (Yap, 2005, p. 6). In conclusion, the author stated that:

> After a year of doing Extreme Programming with a distributed team, the company considers the transition to a global team to be successful. We have demonstrated that Extreme Programming works for a globally distributed group performing around-the-clock continuous development with a shared codebase (p. 7).

Transitioning via Agile Methods

Sepulveda (2003) explained the transition of a small software company interested in the development of new applications beyond the single application it had currently been selling. Due to unforeseen personal issues, the author was forced to manage a newly created software development team from a remote site. In order to facilitate this arrangement, a decision was made to implement a virtual team utilizing agile methods. In the beginning, the team began by using the Extreme Programming practices of pair programming and small releases. In the transition process common challenges to distributed software development were encountered including communication difficulties, lack of trust, and personality differences. In the words of Sepulveda, “remote development changes the mechanism of communication from a tap on the shoulder and face-to-face conversation to a telephone call or ‘instant message’” (p. 3).

In the experience of the Sepulveda (2003), trust was the most difficult area to foster in the distributed environment. In relation to personality differences it was discovered that “in remote settings natural differences are magnified because of lack of face-to-face interaction” (p. 3). In order to alleviate the communication, trust, and personality issues, Scrum-style planning meetings are conducted via conference call each morning.

In addition Sepulveda (2003) suggested the following guidelines for the development of remote teams. First, the use of virtual teams should be thoroughly investigated and the decision should be made upon the basis of necessity rather than novelty. Second, provide the appropriate information and communication technologies such as instant messaging, groupware applications, and shared applications. Third, facilitate trust through timely feedback and clear communication. Fourth, take into consideration cultural context. Fifth, make an attempt to simulate the colocated environment through the information and communication technologies that have been implemented. In conclusion, the author stated, “we actually are more efficient and successful with a virtual team than when we were co-located in the same room” (p. 6).

24-Hour Development Teams

Hogan (2006) presented an experience report on the formation of a 30 person, mixed skill team, distributed across Australia, India, and the United Kingdom. The primary reason for the client’s choice of this configuration “was the flexibility to rapidly scale up the team size to meet the target deadline and to take advantage of potential 24 hour development between teams” (p. 1). Like other experiences with global software development challenges revolving around communication, culture, and technology were encountered.

Initially an attempt was made to conduct daily stand-up meetings to facilitate communication between the remote sites. However, the use of teleconferencing and the large size of the team did not accommodate effective communication. The solution was to establish one-to-one relationships between the managers of each remote site who then served as a representative for the entire team. The manager then coordinated the activities of their local team.

In order to alleviate the problems caused by lack of face-to-face contact, the organization implemented an ambassador exchange program in which each ambassador spent up to a month with each team. The ambassador served a similar role to liaisons or gurus in other distributed software development projects (Carmel & Agarwal, 2001; Yap, 2005). Hogan (2006) reiterated the importance of effective communication and its vital role in the formation of trust between distributed team members. Hogan stated, "trust is critical for effective distributed development" (p. 3).

Hogan (2006) pointed out the following lessons learned: 1) smaller teams were found to be more effective; 2) multidisciplinary teams consisting of project managers, team leads, business analysts, testers, and developers contributed greatly to the success of the project; 3) a method called mirroring was employed whereby the developer, business analyst, and testers in one remote site were matched with their counterpart at another remote site for similar tasks allowing for 24 hour work; 4) a dedicated infrastructure team was assembled to deal with technical situations such as monitoring networks and updating software and databases; 5) various communication tools were needed including instant messaging, voice over IP, video documentation, and remote desktops. In conclusion, it was noted that "distributed teams are extremely reliant on good infrastructure setup for remote development and need reliable network and telecommunications support for communications" (Hogan, 2006, p. 6).

Factors Contributing to Successful Product Creation in Global XP Teams

Layman, Williams, Damian, and Bures (2006) conducted case study research in order to determine what factors allowed global software development Extreme Programming teams to create products successfully. The determination of success included an evaluation of team processes, team productivity, product quality, and customer satisfaction. The case study consisted of a development team located in the Czech Republic, while the project management personnel, project manager, project tracker, and development manager were based in the United States.

The case study followed the development project from initial requirements determination through the completion of the final product. Both qualitative and quantitative data were gathered. Qualitative data were supplied through an informal email message sent to members of the project management located in the United States and to the lead developer in the Czech Republic, as well as face-to-face interviews with both the project management team in the United States and the members of the development team in the Czech Republic. Quantitative data consisted of a count of the number of lines of code, classes, and methods. The qualitative data were analyzed utilizing a grounded theory approach. Because grounded theory does not require any pre-conceived theories, the purpose of this study was not to conduct hypothesis testing, but rather to develop conjectures about the practice of communication within GSD XP teams. As Layman et al. (2006) noted, “these conjectures serve as an initial basis for understanding the challenges facing XP GSD teams and are based on grounded theory observations” (p. 787).

In terms of a globally distributed XP team Layman et al. (2006) presented four conjectures. First, in order establish good decision making and requirements determination it was essential to have a well-defined customer. The authors recommended that this could be done by defining a person up front to play the customer role. Second, to facilitate a two-way communication conduit a key member of one team should be physically located with the other team. A recommendation was to create a member role to be a liaison between the project management team and the development team. Three, commitment, confidence, and a focused development environment can be created by members promptly replying to asynchronous queries. In the absence of face-to-face or synchronous communication it was suggested that an email listserv be implemented. Fourth, process control and plan effectiveness can be improved by making process and product information continuously accessible to the team members. A recommendation was to "use globally-available project management tools to record and monitor the project status on a daily basis" (p. 791). In conclusion, Layman et al. stated that "despite geographic, technical, temporal, and linguistic hurdles, the team was able to create an environment rich with the informal communication that is essential to XP" (p. 792).

Support for Agile Distributed Software Development

Ramesh et al. (2006) conducted a study among three organizations to try and determine whether or not agile distributed software development can be supported. The three companies were located in India, but supported customers located in the United States. Data were collected through semi-structured interviews with a total of 18 software developers, project managers, and customer representatives being interviewed. Grounded theory was used to analyze the qualitative data. The study was conducted over a 9-month period. The result of the study was the identification of five groups of practices that addressed the challenges of agile, distributed development. The practices included: continuously adjusting the process, facilitating knowledge sharing, improving communication, building trust, and trusting by verifying. In conclusion, the authors answered, "yes", to the question of "can distributed software be agile?" This result can occur "when the unique characteristics of both environments are successfully blended" (Ramesh et al., 2006, p. 41).

Summary and Implications of Literature Review

To combat the traditional challenges associated with software development, parallel interest is growing in the use of agile software development methods and global software development through the use of virtual teams as potential solutions. As evidenced by the literature, separately, both agile methods and global software development through the use of virtual teams have distinct benefits for addressing the issue of projects which are over-budget, behind schedule, and inadequate to meet the needs of users. In the past, however, agile software development methods were considered inapplicable in globally distributed environments. More recently, some research suggests that this is no longer the case (e.g., Kircher et al., 2001; Schummer and Schummer 2001; Xiaohu et al., 2004). The combination of agile software development methods and global software development through the use of virtual teams, therefore, poses a potentially rich area of study with potentially significant implications for research practice.

One issue, however, that has not received specific attention is the exploration of the actual configuration of agile teams in globally distributed environments. Thus, this study poses the following research: how can agile teams be successfully configured in globally distributed environments? Team configuration in these global settings is complex. Multiple challenges, as cited in the literature review, exists which are not encountered within colocated teams. For example, issues such as communication, coordination, culture, geographic distance, technology, and logistics pose a significant threat to the successful configuration of globally distributed agile teams. Even the foundation of agile teams is challenged in that these teams have traditionally relied on regular face-to-face interaction. To address this question a theoretical framework drawing upon configurational theory, work design research, team virtualness concepts, and the software agility literature was developed to guide the study.

Theoretical Framework

Configurational Theory

There is a wide ranging base of literature utilizing Configurational Theory (e.g., Dwyer, Richard, Chadwick, 2003; Ebben & Johnson, 2005; Kim, Acito, Rusetski, 2006; Miles & Snow, 1978; Mintzberg, 1979, 1983). Configurations are basically patterns or characteristics that describe an entity. Meyer, Tsui, and Hinings (1993) defined a configuration to “denote any multidimensional constellation of conceptually distinct characteristics that commonly occur together” (p. 1175). Accordingly, configurations can be examined from the standpoint of “multiple levels of analysis, depicting patterns common across individuals, groups, departments, organizations, or networks of organizations” (Meyer et al., 1993, p. 1175). A review of the literature suggested that the majority of studies have occurred at the organizational level of analysis (e.g., Ferratt, Agarwal, Brown, & Moore, 2005; Ketchen, Thomas, & Snow, 1993; Miller 1987; Smith, Shortell, & Saxberg, 1979). The literature also suggested that it is possible to use the configurational approach/perspective without using formal Configurational Theory (Jones, Cline, & Ryan, 2006; Ostroff & Schmitt, 1993). O’Leary and Cummings (2007) mentioned the configurational dimension of virtual teams, but limited its definition to the “arrangement of members across sites independent of the spatial and temporal distances among them”, thus stating that “configuration has to do with the location of team members, not the average distance among them" (p. 439). This study takes a much broader approach to the term configuration. Despite the fact that the unit of analysis has primarily been organizational in nature, Meyer et al. (1993) argued that "the configurational perspective has unrealized potential at other levels as well" (p. 1175). Meyer et al. (1993), therefore, suggested work group design as a possible group level configurational approach.

Work Group Design

The original concepts of work group design have evolved over a significant period of time (Hackman & Oldham, 1980; Hackman & Walton, 1986; Hackman, 1987; Hackman, 1990; Hackman, 2002; Wageman, Hackman, & Lehman, 2005). The latest iteration of the team design model proposed that in order for teams to be effective or successful several enabling conditions must be present. These conditions included: real team, compelling direction, enabling structure, supportive organizational context, and available expert coaching (Hackman, 2002). According to Hackman, real teams must meet three requirements: 1) possess clear boundaries between who is and who is not on the team, 2) members must be collectively interdependent and responsible for the work they produce, and 3) the team must be reasonably stable. Compelling direction addresses the overall purpose of the team. Compelling direction should be challenging, clear, and consequential. Supportive organizational context entails a reward system that provides positive consequences for good work, an educational system that makes adequate training opportunities available, and an information system that provides necessary data for accomplishing assigned tasks. Available, expert coaching provides support for team members in the areas of effort, performance strategy, and knowledge and skill. Because this study seeks to specifically explore team configuration, the decision was made to incorporate the enabling structure condition into the theoretical framework. Enabling structure addresses the key structural elements of task design, core norms of conduct, and team composition. The decision to include only the condition of enabling structure helps to bind the study to the issue of team configuration. It is outside of the scope of this study to try and address all of the enabling conditions proposed by Hackman (2002). Drawing from Hackman's (2002) model and extant literature on virtual teams and agile methods, three key dimensions for configuring globally distributed agile software development teams are proposed: team structure, virtualness, and agility. Brief definitions are provided in Table 2:

Table 2

Brief Definitions of Theoretical Framework Dimensions

Dimension	Definition
Team Structure	The overall design of the team based upon the elements of task design, team composition, and core norms of conduct
Team Virtualness	The extent of virtualness of the team based upon the characteristics of temporal distribution, boundary spanning, lifecycle, and member roles
Team Agility	The extent to which the team aligns with the general values and principles of agile methods and the practices of a specific method

In addition, the implementation of agile methods, global software development, and virtual teams share many of the same challenges. As such, these challenges are included in the model indicating that they impact the process of configuring successful globally distributed agile teams and are briefly defined in Table 3. A visual representation of the theoretical framework is presented in Figure 1 with accompanying written description.

Table 3

Challenges to Globally Distributed Agile Teams

Challenges	Definition
Strategic	Division of work; resistance; scheduling; travel
Cultural	Language; attitudes; values; communication; coordination; conflict management; competency
Communication Official;	informal; mutual knowledge and understanding; relationship building; trust; cohesion
Geographic Distance/time	zones; coordination; control; vendor support
Knowledge Management	Collecting knowledge; sharing knowledge across sites
Project Management	Synchronization; techniques; cultural differences
Technical Networking;	software; compatibility; synchronous and asynchronous information and communication technologies

Team Structure

For the purpose of this study which explicitly deals with team configuration, the components of enabling structure were adopted to form one dimension of the theoretical framework. As stated by Hackman (2002), “it is a fantasy – a tempting and pervasive one, but a fantasy nonetheless – that it is possible to have great teams without the bother of creating enabling team structures” (p. 13). In regard to virtual teams in particular, Powell et al. (2004) commented that many times it appeared as if virtual team design was only an afterthought based upon an examination of the structural characteristics of virtual teams which have been studied up to date. Hackman (2002) stated that “it is a significant managerial task to create an appropriate team structure – let alone to bring a team to life as a real social system – if members are scattered around the region, the country, or the world” (p. 131). Overall, the key to good team design is to determine which structural features are critical and which are unnecessary.

In a distributed project using virtual teams comprised of undergraduate students from universities located in the United States and Germany, Kaiser, Tullar, and McKowen (2000) found that clearly specifying the team structure when designing the team assisted members to better understand their particular role, to comprehend the ultimate goal of the project, and to establish shared norms. Each of these virtual team design issues are addressed by the structural features identified by Hackman (2002) which included: task design, core norms of conduct, and team composition. In sum, Powell et al. (2004) stated, “we believe that investigation of team structure in the virtual environment holds significant promise for research and practice because it represents perhaps the most controllable and influential aspect of virtual team design” (p. 16).

Task Design

Task design deals with the construction of the work itself. According to Powell et al. (2004), significant attention has been paid to the design of virtual team interaction, but much less attention has been given to the design of the work unit itself. The overall goal of a well constructed design is to facilitate collective internal work motivation. In order to reach this goal, task design seeks to provide the team with a meaningful challenge, to allow the team to practice significant autonomy, and to offer the team regular performance feedback (Hackman, 2002).

Meaningfulness. According to Hackman (2002), “the ability to create work that is challenging, complete, and significant – and therefore meaningful to those who carry it out – is one of the major advantages of designing work for teams” (p. 99). One of the strategic challenges of global software development and the use of virtual teams is the division of work across geographically distributed sites (Herbsleb & Moitra, 2001). An element of task design in regard to distributed work is to avoid dividing the work up to the point that the members do not perceive that they are working together on a larger project. If team members lose sight of the importance of the task that they have been assigned there may be a tendency for their motivation levels to decrease. It is important to keep the bigger picture constantly in front of the team members. As noted by Hackman (2002), the deliberate design of work for teams “makes it possible to create tasks that are large and significant” (p. 98). Consequently, when team members sense that they are involved with a meaningful project that has significance to the organization and its customers, they possess greater intrinsic motivation to complete the project.

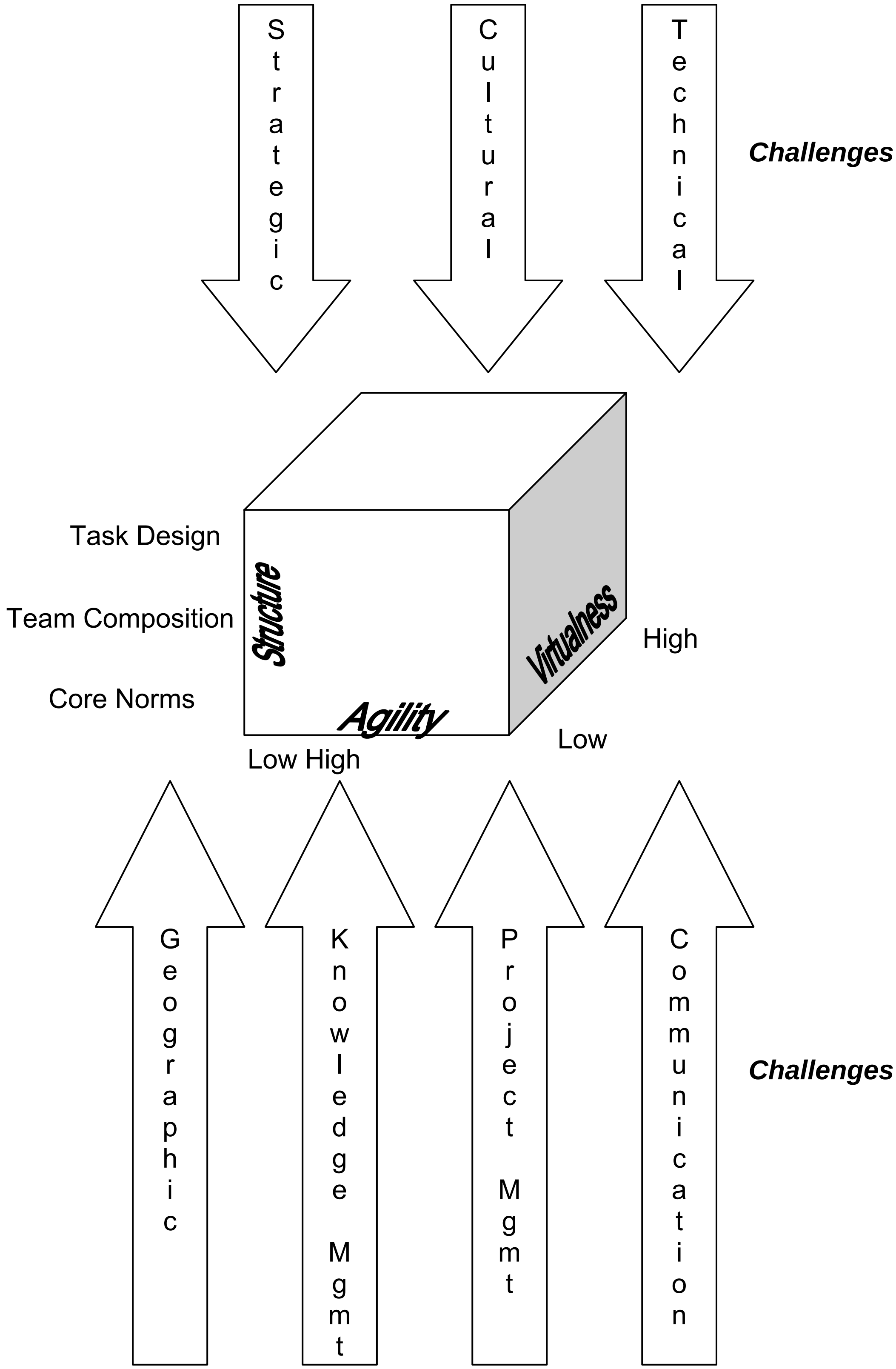

Figure 1. Theoretical framework for globally distributed agile team configuration.

Autonomy. A second aspect of fostering collective internal motivation is to provide "team members a large measure of autonomy to decide how they will use their human and material resources in carrying out the work" (Hackman, 2002, p. 100). Autonomy is also important in distributed work well. According to Herbsleb and Moitra (2001), in global software development, ideally the teams at each site would to a large degree work independently while continuing to communicate with other sites to keep the entire project in focus. Battin et al. (2001) recommended distributing "entire things for the entire lifecycle" as a way to reduce communication difficulties, but this idea can also be implemented as a way to provide additional autonomy to distributed teams. In relation to virtual teams, Suchan and Hayzak (2001) emphasized that those individuals chosen for the team must be independent and self-directing, but at the same time be able to work interdependently with other team members. At the least, the team should have the responsibility to oversee its own work processes (Hackman, 2002).

Performance Feedback. Finally, collective internal motivation is increased by regular performance feedback. A team cannot evaluate itself and learn unless it has some data about how it is doing. Subsequently, without learning, there is no opportunity for improvement (Hackman, 2002). One of the issues regarding performance feedback in globally distributed teams is their ad hoc nature. Many times teams are formed for short projects and are disbanded before the feedback loop can be completed. However, it should be the goal of the project leader to attempt to provide as much feedback as possible to the team before it is dissolved.

Team Composition

Team composition addresses the elements of size, mix, interpersonal skills, and task-related knowledge and skill. According to Hackman (2002) organizations tend to make three common assumptions in regard to the compositions of teams: 1) they assume the "bigger the better" and often place too many members on the team; 2) they assume that similar people will work better together and form teams which are too homogenous; and 3) they assume that individuals already know how to work in a group and fail to consider the interpersonal skills of prospective members. However, the one mistake that is less likely to be made is to ensure that team members have the appropriate task-related knowledge and skill.

Size. In regard to the size of the team, Hackman (2002) advocated having as few team members as possible to accomplish the task, in fact, "a team may function better when it has slightly fewer members than the task actually requires" (p. 118). Hackman cited evidence that while initial productivity does actual increase as size increases, it eventually levels off, and actually begins to decrease for very large groups. In the end, "when group size becomes very large, the problems generated far outweigh the incremental resources brought by the additional members" (Hackman, 2002, p. 117). With regard to global software development, Herbsleb and Mockus (2003) found that the number of people working on a distributed project directly related to delays in the project. Determining the size of the team, therefore, is dependent on the size of the task. In relation to team size, Powell et al. (2004) stated that in their review of the literature on virtual teams, no specific study had been conducted that explicitly examined virtual team size as a variable during the team design phase.

Mix. Hackman (2002) suggested that “a well-composed team strikes a balance between having members who are too similar to one another on the one hand and too different on the other” (p. 122). He cited evidence that there is little proof that homogenous teams outperform heterogeneous teams and that always working in a homogenous team is less likely to facilitate the learning process of the members. In globally distributed teams it is reasonable to expect that the teams will be much less likely to be homogenous due to the fact that the sites are geographically distributed. This is one reason why cultural and communication issues pose such a challenge to globally distributed teams.

Powell et al. (2004) suggested that there is a need for research which examines the characteristics of successful virtual teams and what managers should be looking for when choosing virtual team members. Powell et al. also pointed out that little research has been conducted which examines the personalities of individuals who are more amenable to working in virtual teams. Balijepally, Mahapatra, and Nerur (2006), however, examined the personality characteristics of agile software development teams which may provide additional insight when configuring globally distributed agile teams. In sum, Hackman (2002) argued that the “key to having a good mix of members is to balance carefully between too much similarity and too many differences” (p. 124).

An element related to the mix of virtual team members should also consider what Crampton (2001) called “mutual knowledge”. Mutual knowledge was defined as that knowledge that team members communicate with each other and which they know they share in common. For instance, a team comprised of members from the same organization that are distributed across the United States, would possess considerably more mutual knowledge than a team comprised of members from multiple organizations distributed across sites located in different countries. Thus, when designing the mix of a team, it is important to consider what knowledge would be shared or is "mutual" and what is not and consider measures for distributing this knowledge across the teams.

Interpersonal skills. Hackman (2002) stated that “some people just are not cut out to be team players” (p. 125). In his opinion, the ideal team would be composed of members who meet a predefined standard for interpersonal skills. This not only holds true in traditional teams, but also in virtual teams. Suchan and Hayzak (2001) argued that virtual team members must also possess excellent interpersonal and conflict management skills. The importance of addressing interpersonal skills within the context of team structure cannot be overstated. Hackman (2002) argued that in an appropriately structured team the number of interpersonal conflicts would be less than in a team for which task design, core norms, and composition were given little or no thought.

Core Norms

Core norms of conduct specify the acceptable and unacceptable behaviors agreed upon within the team (Hackman, 2002). The establishment of a shared set of norms which direct the individual and corporate behavior of virtual teams has been cited as an import element of global virtual team design (Sarker et al., 2001; Suchan &

Hayzak, 2001). During the formation of the team there may be much ambiguity about member roles, overall goals, and the rules which will govern the actions of the team. As such, the team leader or project manager will be called upon to begin the process of defining these areas. In regard to norms, it is important that each team member positively internalizes this set of rules and in essence "buys-in" to their use (Sarker et al., 2001).

Hackman (2002) identified three ways that norms are formulated within a group: 1) they are brought in by the individual members of the team; 2) they evolve over time as team members try out certain behaviors; and 3) they are deliberately created as a part of the team structure. Hackman suggested that norms should be categorized as primary and secondary.

Primary Norms. Primary norms are fundamental and outward looking. As such, “members should take an active, rather than reactive, stance toward the environment in which the team operates, continuously scanning the environment and inventing or adjusting their strategies accordingly” (Hackman, 2002, p. 106). Secondly, primary norms establish those things that must “always” be done and those things that must “never” be done within the team. It is important that these two core norms be deliberately created as a part of the team’s structure. Hackman cited two reasons why these two core norms should be explicitly established:

> 1) our disposition to react to whatever captures our attention and demands a response, rather than to actively scan our environment for less obvious problems and opportunities that may call for nonstandard actions (p. 108)

2) our understandable impulse to have harmonious interactions with others, to be approved rather than rejected by our teammates, and generally to keep anxieties as low as possible (p. 109)

Secondary Norms. Secondary norms constitute additional norms that address those behaviors which the members deem as important enough to regulate. These norms may address such issues such as punctuality, participation, communication, and conflict management (Hackman, 2002). Due to the cultural diversity, language barriers, and communication differences in globally distributed teams, these secondary norms can help alleviate conflicts within the team. As a part of the secondary norms common goals and strategies should also be established (Kaiser et al., 2000; Kayworth & Leidner, 2001; Malhotra, Majchrzak, Carman, & Lott, 2001; Suchan & Hayzak, 2001).

Team Virtualness

A second dimension of the theoretical framework for exploring the question of successful configuration of globally distributed agile teams was premised on the work of Bell and Kozlowski (2002) regarding the characteristics who differentiate the "virtualness" of a team. Bell and Kozlowski argued that not only are traditional, colocated teams distinct from virtual, distributed teams, but also that virtual teams may be differentiated from each other as well.

Bell and Kozlowski (2002) first identified two primary differences between traditional and virtual teams: (1) spatial distance, and (2) information, data, and personal communication. Spatial distance refers to the fact that virtual teams are separated by some measure of physical space, that is, they are distributed. Traditional teams on the other hand work in very close physical proximity, they are proximal. Bell and Kozlowski stated that “the specific distance that separates team members is not as important as the effect this spatial separation has on how team members interact” (p. 22). Interaction between traditional teams is typically face-to-face, while virtual team members interact via various information and communication technologies. In sum, “it is the absence of this proximal, face-to-face interaction between members of virtual teams that makes them virtual and distinguishes them from more traditional teams” (p. 22). Virtual and traditional teams are also differentiated by the way information, data, and personal communication was handled. As mentioned above, the main conduit of communication is technology. The use of this mediating technology was not nearly as crucial in traditional teams since most of the time they have the ability to meet face-to-face.

Second, Bell and Kozlowski (2002) proposed that not only was there a distinct difference between virtual and traditional teams, but that there were also distinctions between types of virtual teams. In order to show these distinctions, they developed a typology for the classifying virtual teams. The concepts for the development of typologies are grounded in Configurational Theory (Miles & Snow, 1978; Mintzberg, 1979, 1983). Bell and Kozlowski theorized that virtual teams can be placed on a continuum where one extreme represents the “ideal” type of virtual team while the opposite extreme of the continuum represents a type of virtual team that closely resembles a traditional, colocated team. The “ideal type is defined as follows: “it is distributed across time, spans numerous functional, organizational, and cultural boundaries; is short-lived; and is composed of members who each possess multiple roles in different virtual teams" (p. 28). The more conventional type of virtual team is defined this way: "is temporally entrained, has less permeable boundaries, has a continuous lifecycle, and is composed of members who have singular roles" (p. 28-29). This typology, therefore, serves to characterize teams as more or less virtual based on a combination of characteristics which included temporal distribution, boundary spanning, lifecycle, and member roles.

Temporal Distribution

According to Bell and Kozlowski (2002) temporal distribution denoted that a virtual team was distributed across time. Virtual team members may be colocated in time, separated by only a few hours, or separated by many hours. Additionally, members may also be temporally synchronized, e.g., they are located in different time zones, but are all still working off of the same time reference. Bell and Kozlowski provided the example of space shuttle ground operators who are distributed across the world to illustrate temporal synchronization. The technology employed by the team also plays a role in the determination temporal distribution. Synchronous information and communication technologies allow for real time interaction between team members who are separated even by time zone differences. On the other hand, asynchronous technologies create greater temporal distribution even for colocated teams. The physical location and the technology utilized determine whether the team is truly distributed across time or entrained by real time. Bell and Kozlowski pointed out that the decision to interact in distributed as opposed to real time is to a large degree determined by the complexity of the task. More complex tasks require more real time interaction, whereas less complex tasks can be accomplished in distributed time.

Boundary Spanning

Boundary spanning referred to the fact that virtual teams not only cross the bounds of space and time, but also functional, organizational, and cultural boundaries as well (Bell & Kozlowski, 2002). Boundary spanning allows virtual teams to exhibit the characteristics of adaptability, flexibility, and responsiveness that are desired by many organizations. Additionally, boundary spanning dovetails with the global software development in that both allow for drawing from a large pool of qualified individuals from around the world. The desire for a 24-hour time clock for global software development is also a benefit afforded by boundary spanning (Bell & Kozlowski, 2002; Carmel & Agarwal, 2001; Damian & Moitra, 2006; Herbsleb & Moitra, 2001). However, as seen with global software development, spanning cultural boundaries also involves multiple challenges including language, tradition, technology, and values. Similarly, according to Bell and Kozlowski (2002), the permeability of these multiple boundaries is very much contingent on the team tasks. When tasks are more complex, boundaries tend to be more rigid, while less complex tasks allow the boundaries to be more flexible.

Lifecycle

Bell and Kozlowski (2002) contended that virtual teams experience variable lifecycles. Due to their ad hoc, flexible nature, virtual teams can be formed quickly, but may only be in existence for a short period of time depending upon the project to which they are assigned. As such, virtual teams do not follow the traditional life cycle that occurs in traditional teams. The “ideal” type of virtual team possesses a discrete lifecycle, i.e., they are formed and deployed quickly, accomplish the task, disband and are redeployed. In addition, membership is very dynamic. Additionally, depending on the nature of the task, different virtual teams may go through different life cycle stages than other virtual teams. As with temporal distribution and boundary spanning, team lifecycle is also dependent on task design. As tasks become more complex there is a need for a more continuous lifecycle. On the other hand, less complex tasks allow for a more discrete lifecycle. When tasks are complex it is more difficult to introduce new members into the team and a higher level of cohesion and collaboration is needed (Bell & Kozlowski, 2002)

Member Roles

Finally, virtual teams provided for the selection of members from a substantial pool of workers with a diverse set of skills and allow for greater organizational flexibility (Bell & Kozlowski 2002). Ideally members participate in multiple roles within multiple teams. When tasks are less complex “the roles of virtual team members are more interchangeable” (p. 35). However, as tasks become more complex member roles become more defined.

Each of these characteristics served as the basis for multiple virtual team configurations which lie between the two extremes defined by Bell and Kozlowski (2002). Bell and Kozlowski argued that “virtual teams need to adopt different characteristics to successfully operate within the constraints that are imposed by the complexity of their collective task” (p. 16). Each of these characteristics of virtualness is directly affected by the team task, which is one of the reasons it is so important to carefully consider its design (Hackman, 2002). According to the basic principles behind agile methods, current agile software development teams exemplify the characteristics of a traditional, colocated team configuration. With the move toward global software development, the formation of agile development teams within a globally distributed environment calls for an alternative and more agile configuration. Based on the proposed theoretical framework, the measure of team virtualness, “low” or “high”, can be investigated by examining the characteristics of temporal distribution, boundary spanning, lifecycle, and member roles.

Team Agility

The third dimension of the theoretical framework is team agility which may be defined as how closely the team aligns with the general values and principles of agile methods as well as with practices of a specific method. These values, principles, and practices are drawn from the Agile Manifesto as well as from specific agile methods and help to distinguish agile teams from more traditional plan-driven teams. For instance, the Agile Manifesto ascribes the following values to agile teams: individuals and interactions, working software, customer collaboration, and responding to change. Plan-driven teams, on the other hand, emphasize processes and tools, comprehensive documentation, contract negotiation, and following a plan (Fowler & Highsmith, 2001). Extreme Programming (XP) represents an example of a highly popular agile methodology comprised of twelve specific practices including system metaphor, planning game, small releases, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, forty-hour week, on-site customer, and coding standards.

Proponents of agile methods have consistently argued that in order for agile methods to be successful and bring the best benefits, they must be implemented as a whole (Beck, 2000; Schwaber & Beedle, 2002). However, some recent research suggests that agile methods can be tailored and that specific values, principles, and practices can be chosen and used with benefit (e.g., Kircher et al., 2001; Fitzgerald et al., 2006; Schummer & Schummer, 2001; Xiaohu et al., 2004). According to the proposed theoretical framework, teams with “high” agility are adhering to a greater degree on the general values of agile methods and to a greater number of specific practices of their chosen agile methodology. Conversely, teams with “low” agility have chosen to adopt a lesser number abandoned most of the general values and specific practices.

Challenges to Successful Configuration

As indicated from the review of the literature, many of the challenges associated with the individual use of agile methods, global software development, and virtual teams overlap and represent possible hindrances to the successful configuration of globally distributed agile teams. Because of this reason, the following common or overlapping challenges have been included in the theoretical framework to indicate the potential impact they may have on successfully configuring globally distributed agile teams. Specifically, the following challenges have been included: strategic, cultural, communication, geographic, knowledge management, project management, and technical.

Defining Successful Team Configuration

In terms of defining a successful team configuration, successful was operationalized very broadly as a favorable or desired outcome (Landau, 1977). In this study, successful was related to the term optimal defined as most favorable, best, most desirable, or satisfactory (Landau, 1977; Mish, 2003). Thus, successful was be defined in terms of each member's perception of "successful" based upon their personal experience working on the team and indicated by the self-reported data they provided rather than strictly operationalized by a specific measurement instrument. This approach was be taken due to the fact that the terms "success" and "successful" come with considerable baggage and there is a distinct line of research that deals with defining and measuring team success which was deemed outside the scope of this particular study. The goal of this study was to explore what constitutes a successful configuration rather than specifically measuring team success. Of course, there is a fine line that is acknowledged between these two purposes. Within the interview protocol participants were asked to comment on the question of what constituted a successful configuration to them, e.g., "*In your opinion, what makes a globally distributed agile team configuration successful?"* based upon the three dimensions of team structure, virtualness, and agility; the emphasis here on the term "configuration". Thus, the central question was, "based upon your experience working on this type of team, what is the 'best' way to configure it in terms of its structure, virtualness, and agility"?

CHAPTER 3

RESEARCH METHOD

This chapter describes the research methodology that was used to conduct an exploratory study for the purpose of building theory in the area of successful configuration of agile teams in globally distributed environments that was grounded in the data and integrated with prior research. The study utilized an embedded multiple-case research design based upon theoretical sampling through purposeful, convenience, and snowball sampling techniques for determining the cases to be explored. Due to the number of cases the study was able to employ both literal replication logic and theoretical replication logic to explore the similarities and differences between the cases. Sections included in this chapter cover the case study as a research approach, the use of case study research in the information systems field, and the steps for theory building using a case study method as proposed by Eisenhardt (1989). Based upon these steps, where applicable, specific details of the study are discussed including the selection of cases and sample size, the unit of analysis, the data collection procedures, and the data analysis methods.

The Case Study as a Research Approach

Qualitative research has been greatly influenced by the work of Glaser and Strauss (1967) in the area of grounded theory, Yin (2003) in regard to case study research, and Miles and Huberman (1994) in the field of data analysis. Eisenhardt (1989) utilized this culmination of knowledge to devise a process for building theory based on case study research and subsequently Pare (2000b) extended the work of

Eisenhardt. A case study as defined by Yin (2003) represents “an empirical inquiry that investigates a contemporary phenomenon within its real-life context especially when the boundaries between phenomenon and context are not clearly evident” (p. 13). Similarly, Orum, Feagin, and Sjoberg (1991) defined a case study as “an in-depth, multifaceted investigation, using qualitative research methods, of a single social phenomenon” (p. 2). In general the decision to employ case study research should be based on the following criteria: 1) the study involves a broad and complex phenomenon, 2) the study takes place in its natural setting, 3) the study seeks to answer “how” and “why” questions in order to gain a better understanding of the nature and complexity of the subject, and (4) the study addresses a topic for which there is a limited amount of previous research (Benbasat et al., 1987; Bonoma, 1985; Yin, 2003). Put succinctly, case research is “useful when a phenomenon is broad and complex, where the existing body of knowledge is insufficient to permit the posing of causal questions, and when a phenomenon cannot be studied outside the context in which it naturally occurs” (Bonoma, 1985, p. 207).

The Use of Case Study Research in Information Systems

The use of case study research in the information systems (IS) discipline has gained increasing legitimacy over the last two decades and is considered well-suited to IS research. According to Dube (2004), “case study research is now accepted as a valid research strategy within the IS community” (p. 234). Factors contributing to this acceptance and respect of case research include the following: the overall shift from technical to organizational issues, the use of “real-life” environments in which to conduct research on the rapidly changing nature of IT and organizations, the ability to conduct holistic investigation to better understand the complex relationship between organizations, technologies, and people, the opportunity to examine cases in-depth in an attempt to identify opportunities and challenges, and the use of case research for exploration, hypothesis generation, and testing (Dube & Pare, 2003). Overall, the use of case research in IS has been adequately examined (e.g., Alavi & Carlson, 1992; Benbasat et al., 1987; Dube & Pare, 2003; Klein & Myers, 1999; Orlikowski & Baroudi, 1991; Lee, 1989). In fact, the case research conducted by Markus (1983) into management information system implementation is considered an exemplar of case research within the IS field (Lee, 1989). In sum, Benbasat et al. (1987) stated that “the case research strategy is well-suited to capturing the knowledge of practitioners and developing theories from it” (p. 370).

The Use of Case Study Research to Build Theory

The overall goal of this study was to build theory in regard to globally distributed agile team configuration and to formulate propositions regarding how this theoretical base helps exploration of a successful configuration of this type of team. Eisenhardt (1989) represents one of the seminal articles for building theory through the use of case study research. The steps proposed by Eisenhardt have served as a framework for subsequent studies (e.g., Pare, 2000a) and have been examined in detail as a guide for conducting theory building case research (Guy & Pare, 1997; Pare, 2002b, 2004). This study followed the guidelines proposed by Eisenhardt (1989) where applicable as shown in Table 4.

Table 4

Process of Building Theory from Case Study Research

Step	Activity	Reason
1. Getting Started	Definition of research question	Focuses efforts
	Possibly *a priori* constructs	Provides better grounding of construct measures
	Neither theory nor hypotheses	Retains theoretical flexibility
2. Selecting Cases	Specified population	Constrains extraneous variation and sharpens external validity
	Theoretical sampling	Focuses efforts on theoretically useful cases
3. Crafting Instruments	Multiple data collection methods	Strengthens grounding of theory by triangulation of evidence
4. Entering the Field	Overlap collection and analysis	Speeds analyses and reveals helpful adjustments to data collection
	Flexible data collection methods	Investigator may take advantage of emergent themes and unique case features
5. Analyzing Data	Within-case analysis	Gains familiarity with data and preliminary theory generation
	Cross-case analysis	Look beyond initial impressions and see evidence through multiple lenses
6. Shaping Propositions	Iterative tabulation of evidence	Sharpens construct definition, validity, and measurability
	Replication logic across cases	Confirms, extends, and sharpens theory
	"Why" behind relationships	Builds internal validity
7. Enfolding Literature	Conflicting literature	Builds internal validity, raises theoretical level, and sharpens construct definitions
	Similar literature	Sharpens generalizability, improves construct definition, and raises theoretical level
8. Reaching Closure	Theoretical saturation	Ends process when marginal improvement becomes small

Note: Adapted from Eisenhardt (1989).

Research Question and A Priori Constructs

Definition of Research Question

According to Eisenhardt (1989) “an initial definition of the research question, in at least broad terms, is important in building theory from case studies” (p. 536). This sentiment is shared by Yin (2003) as well. Not only does defining the initial research question assist the researcher to gain a clearer focus it can also help the researcher from becoming too overwhelmed by the amount of data collected (Eisenhardt, 1989). As such, this study seeks to contribute to the IS discipline by exploring the question of how agile teams can be successfully configured in globally distributed environments.

In order to truly understand the configuration of team structure, team agility, and team virtualness, and how multiple challenges impact each of these dimensions, the researcher needed to examine each of these components within the actual team setting in its natural environment. The research question itself sought to answer “how can agile teams be successfully configured in a globally distributed environment” based upon the dimensions of team structure, agility, and virtualness, thus forming a complex phenomenon between the dimensions of the theoretical framework. In addition, as shown by the review of related literature, previous studies investigating the use of agile teams in globally distributed environments is relatively small. Finally, no theoretical framework exists which specifically addressed the successful configuration of globally distributed agile teams. Within the context of the wealth of accumulated knowledge in the qualitative research tradition and based upon the reasons provided above which were consistent with general case study principles as well as specific guidelines in the IS discipline the choice of a case study approach was chosen as the research method.

Development of A Priori Constructs

One of the most important aspects of building theory from case study research is that the research should begin “as close as possible to the ideal of no theory under consideration and no hypotheses to test” (Eisenhardt, 1989, p. 536). However, Eisenhardt (1989) admitted that “it is impossible to achieve this ideal of a clean theoretical state”, thus suggesting that the “a priori specification of constructs can also help the initial design of theory-building research” (p. 536). Based upon work group design research, and a review of the related literature, this study proposed an initial theoretical framework which included the following dimensions: team structure, virtualness, and agility. The framework also included the pertinent challenges that may hinder the successful configuration of the team. This theoretical framework was discussed in detail and a visual representation was presented in Chapter 2.

According to Yin (2003), the “role of theory development, prior to the conduct of any data collection, is one point of difference between case studies and related methods such as ethnography and grounded theory” (p. 28). The purpose of this framework was to provide a starting point or blueprint for the research and to give an overall structure for exploring the research question while retaining theoretical flexibility (Eisenhardt, 1989; Yin 2003). It is important to note that although a priori constructs are helpful in the initial phase of the study, these dimensions are at best tentative, and “no construct is guaranteed a place in the resultant theory, no matter how well it is measured” (Eisenhardt, 1989, p. 536). Thus, based upon the analysis of the data, the theoretical framework went through an iterative process until the framework accurately reflected the findings of the data.

Neither Theory nor Hypotheses

Following the suggestions of Eisenhardt (1989), an initial research questions was posed to guide the direction of the study and an *a priori* theoretical framework was developed based upon the extant literature. However, no *a priori* hypotheses, propositions, or relationships were formulated. This, too, is consistent with the suggestions of Eisenhardt (1989) in that,

> . . . investigators should formulate a research problem and possibly some potentially important variables, with some reference to extant literature. However, they should avoid thinking about specific relationships between variables and theories as much as possible, especially at the outset of the process" (p. 536).

The formulation of propositions was generated through an iterative process of analyzing the data and will be presented within the discussion of the cross-case analysis later in this chapter.

Case Selection and Sample Size

Specified Population

In addition to defining the initial research question, the consideration of case selection is an equally important element of theory-building case study research (Eisenhardt, 1989; Yin, 2003). Although consideration of a specific population and subsequent selection of random samples from that population may be utilized in case study research, more common is the use of theoretical sampling as opposed to the random, statistical sampling found in traditional hypothesis-testing research conducted in the quantitative methodologies. Eisenhardt (1989) asserted that in regard to theory-building case study research, “random selection is neither necessary, nor even preferable” and that “the sampling of cases from the chosen population is unusual when building theory from case studies” (p. 537).

Theoretical, Not Random, Sampling

When deciding on the selection of cases it is important to consider the overall purpose of the study. In general, case study research design addresses two primary considerations: (1) will the study employ a single- or multiple-case design, and (2) is each case holistic or embedded. Yin (2003) argued that a single-case is appropriate when the case study represents a critical case, an extreme or unique case, a typical case, a revelatory case, or a longitudinal case. Multiple-case designs, on the other hand, involve more than one case, and are appropriate when the objective is either to identify similarities or differences between two or more cases. Because of this fact, multiple-case designs are often considered more rigorous and generalizable than single-case designs. In multiple-case designs the decision about how many cases to include should be considered carefully. Yin (2003) stated that “every case should serve a specific purpose within the overall scope of inquiry” (p. 47).

In regard to holistic or embedded, holistic refers to the fact that each case stands alone, whereas, embedded indicates that the case is part of a larger unit, as is the situation when studying teams which are part of a larger organization. Because cases can be embedded, specifically identifying the unit of analysis is a key component. By doing so, the boundaries of the study can be better limited (Pare, 2004). According to

Yin (2003), the unit of analysis must be related to the definition of the initial research question.

The use of theoretical sampling can help when making this choice. Theoretical sampling entails the selection of cases “to replicate previous cases or extend emergent theory, or they may be chosen to fill theoretical categories and provide examples of polar types” (Eisenhardt, 1989, p. 537). When addressing theoretical sampling, Yin (2003) differentiated between selecting cases based on either literal replication or theoretical replication logic. Literal replication logic is based on the rationale of selecting cases based on the expectation of similar results. According to Eisenhardt (1989) the overall goal of theoretical sampling in regard to literal replication logic is to “choose cases which are likely to replicate or extend emergent theory” (Eisenhardt, 1989, p. 537). Conversely, theoretical replication logic is based on the rationale of selecting cases based on the expectation of contrasting results (Yin, 2003). Regardless of the replication logic chosen, a rich theoretical framework is an important step in the design process because it “later becomes the vehicle for generalizing to new cases” (Yin, 2003, p. 48). Yin argued that it is important to understand that case study research involves analytic rather than statistical generalization. Analytic generalization involves the comparison of empirical results with theory that has already been developed. Thus, replication may be claimed “if two or more cases are shown to support the same theory” (Yin, 2003, p. 33). In sum, the main point Yin attempted to make was that the case study researcher should,

> try to aim toward analytic generalization in doing case studies, and you should avoid thinking in terms as ‘sample of cases’ or the ‘small sample sizes of cases’

> as if a single case study were like a single respondent in a survey or a single experiment in an experiment (p. 33).

Within the context of literal replication logic the actual number of cases selected is primarily dependant upon the discretion and judgment of the researcher (Yin, 2003). Another consideration is the amount of certainty the researcher desires to have in regard to the results of the multiple-case design (Pare, 2002b). More specifically, Eisenhardt (1989) suggested a number somewhere between four and ten cases as an appropriate number in order to generate theory; whereas, the selection of more than ten cases can lead to a difficulty in managing the volume of data and complexity. Based upon these recommendations from the literature the rationale for selecting the participating organizations and teams is provided in the following sections.

Identification and Selection of Organizations

The identification and selection of organizations was based upon purposeful sampling using convenience and snowball sampling techniques (Patton, 1987, 2002). Criteria for organization selection included the following: (1) globally distributed sites, (2) teams using an agile methodology in some form, and (3) a willingness to participate. One of my dissertation co-chairs provided me with contact names at several potential organizations from the College of Business Administration advisory board. Each of these individuals was contacted via email in regard to the purpose of the research study and an explanation of the organizational and team selection criteria as well as procedures for protecting anonymity and confidentiality was included. I also had personal contacts with additional organizations and I contacted those individuals via email to inquire about the possibility of their organization participating in the study. Follow-up phone calls were made to these individuals and additional names were provided for me to contact.

Ultimately, three organizations in transportation-related industries were selected. Two of the organizations were U.S. based, multi-national with employees located across the world. One organization was U.S. based and utilized offshore contractors as part of its team structure. The decision to include multiple organizations was made in order to increase the likelihood of multiple team configurations that would allow for sufficient examination of the dimensions of the theoretical framework and to increase internal validity of the study as the organizations were similar in that they were in related industries, they were globally in nature, and they were utilizing globally distributed agile teams. However, there were differences in overall size, organizational structure, and extent to which the organization supported the use of agile methods. More detailed description of each organization is provided in Chapter 4.

Identification and Selection of Cases

According to Yin (2003), the unit of analysis should be directly tied to the research question and the selection of cases. For this study the unit of analysis was the team itself which was embedded within the organization. Thus, after receiving organizational approval to proceed and the required non-disclosure agreements were signed the identification and selection of the cases was conducted. Following the recommendation of Eisenhardt (1989) and Yin (2003) cases were selected based upon theoretical sampling, not sampling logic, utilizing purposeful and snowball sampling techniques. As indicated by Patton (2002), purposeful sampling “focuses on selecting information-rich cases whose study will illuminate the questions under study” (p. 230) and the “specific type and number of cases selected depends on the study purpose and resources” (p. 243). Information-rich cases were defined as “those from which one can learn a great deal about issues of central importance to the purpose of the research” (p. 46). The specific criteria utilized were as follows: 1) the team was currently utilizing an agile methodology either formally or informally and 2) the team consisted of members who were distributed across global sites.

A snowball sampling technique was utilized to identify the teams because it facilitated finding such information-rich cases and major informants from “sampling people who know people who know what cases are information rich, that is, good examples for the study, good interview participants” (Patton, 2003, p. 243). Starting with the contact persons provided by the upper management representative from each organization I started to contact these individuals who in some cases provided the names of other individuals until I reached the specific persons who could assist me Final contact persons were project managers from each individual team. Follow-up phone calls were made to each project manager to discuss the arrangement and means for contacting individual team members to set up the telephone interviews. It was agreed that the project manager would supply a list of names and contact information for each team member. I then proceeded to contact the members via email to describe the purpose of the study and to illicit their participation. I then worked with each individual member to set up a date and time for the interview. In one team the supervisor assisted me with supplying recommended times for interviewing each member on that portion of the team.

As indicated by Patton (2002), “there are no rules for sample size in qualitative inquiry” it is dependent upon multiple factors such as “what you want to know, the purpose of the inquiry, what’s at stake, what will be useful, what will have credibility, and what can be done with available time and resources” (p. 244). For this study a total of five teams were included from the three participating organizations providing the appropriate number of cases suggested by Eisenhardt (1989) and Yin (2003) for facilitating both literal and theoretical replication. Across all teams this study included 37 individual team members serving in multiple roles including software developers, project managers, architects, technical leads, business analysts, quality assurance analysts, and test analysts located in various parts of the world including Australia, Brazil, India, Mexico, Poland, and the United States.

In sum, this study utilized an embedded multiple-case research design based upon theoretical sampling through purposeful and snowball sampling techniques for determining the cases to be explored. Due to the number of cases the study was able to employ both literal replication logic and theoretical replication logic to explore the similarities and differences between the cases. By examining existing agile teams in globally distributed environments it was hoped that certain patterns and themes would emerge. The intent was to build a theory which explored the dimensions contributing to a successful team configuration. A result of this study may be the establishment of a set of best practices applicable to other globally distributed agile teams in other organizations.

Data Collection

Data Collection Methods

Interviews are typically the primary source for collecting data in case study research (Pare, 2004; Yin, 2003). In general, interviews may take the form of unstructured, semi-structured, or structured. Unstructured interviews are most appropriate when the interviewer has little knowledge about the subject and is using the interview as a means to learn more about it. In unstructured interviews the interviewer typically does not have a set of predefined questions. In semi-structured interviews a set of predefined questions is used, but since the interviewer does not know exactly how the interviewee will respond additional questions may be formulated as the interview proceeds. According to Berg (2004) a semi-structured interview “involves the implementation of a number of predetermined questions and special topics”; subsequently, the questions are “typically asked of each interviewee in a systematic and consistent order, but the interviewers are allowed to digress; that is, interviewers are permitted (in fact, expected) to probe far beyond the answers to their prepared standardized questions” (p. 81). Thus, semi-structured interviews are effective at both supplying the interviewer with the needed information through predefined questions and also providing flexibility to the interviewee to offer additional information. Finally, structured interviews consist of a set of predefined questions with a set of limited responses to those questions. Structured interviews provide much less flexibility than semi-structured interviews (Pare, 2004). Other sources of data included documentation, archival records, direct observation, participant-observation, and physical artifacts.

The primary data collection method for this study consisted of individual, semi-structured telephone interviews conducted over the time period of June 2007 to April 2008. The interviews were initially scheduled for one hour, but actual interview times varied from approximately 25 minutes to one hour forty-five minutes. The total interview time for all interviews was approximately 30 hours while the transcription time was approximately 63 hours. The transcriptions of the interviews contained approximately 500 pages, 27,000 words, having a mean length of 4,000 words per interview. The process of data collection started with an initial conversation with the project manager of each team. The project manager assisted me in determining which individual team members to contact. In some cases, especially in smaller teams contacting all the members was suggested. In regard to the larger teams, the project manager attempted to direct me to a good mixture of roles and to those members with a stronger fluency in English for those located in other countries. This process followed the concept of purposeful sampling to steer me toward key informants and information-richness (Patton, 2002). Of course every member on the team was a potential key informant but due to time constraints and the busyness of each team member's schedule it was not possible to interview every member on every team. Additionally, some members did not respond after several email attempts. In the situation of two cases the offshore contractors were not available for interviews due to non-disclosure reasons. After receiving the list of members and their contact information, an email was sent to each member explaining the research study and asking for their participation.

The interviews were guided by an interview protocol and a case study protocol which were developed prior to conducting the interviews (see Appendix B and Appendix

D respectively). The case study protocol serves as a primary way of strengthening the case study reliability and is vital in a multiple-case study (Yin, 2003; Pare 2002b). The case study protocol typically contains the following sections: 1) an overview of the case study project, 2) field procedures which will be followed, 3) case study questions, and 4) a guide for the case study report. Chapter 1 of this study presents an overview of the case study project. Chapter 3 outlines field procedures to be followed in terms of selection of cases, data collection, case study protocol steps, and data analysis. Appendix B presents the interview protocol which contains the specific questions to be asked of participants. The case study protocol is presented in Appendix D which consists of potential sources of data for each of the constructs as well as the associated questions of the case.

As suggested by Yin (2003) the case study protocol can help increase the reliability of the study by ensuring that each interview is conducted in the same way. In addition to the case study protocol presented in Appendix D, case study protocol steps were developed prior to the study to ensure that all interviews followed the same format as presented in Table 5. If possible the informed consent document, presented in Appendix A, was emailed to the participant for them to review in advance. The participant was then called, the informed consent document was read if not reviewed previously, and the participant was asked whether they understood the informed consent and if they had any questions in regard to the study in terms of anonymity and confidentiality. It was explained that the participant could stop the interview at any time. Verbal permission was received by each participant to digitally record the interview. The purpose of the study was then briefly explained and a brief definition of configuration was provided to the participant.

Table 5

Case Study Protocol Steps

Steps in the Case Study Protocol
1. Email informed consent script if possible
2. Turn on phone recording controller
3. Turn on digital recorder
4. Make phone call
5. Read informed consent script (if not previously reviewed), explain purpose of the study, and define configuration *"In this study, configurations are defined as patterns or characteristics that describe an entity. The specific dimensions being explored include team structure, team agility, and team virtualness."*
6. Press record on digital recorder
7. Begin with demographic questions
8. Proceed to questions from the interview protocol
9. Code audio recording with identification number
10. Transfer digital recording to password-protected computer
11. Transcribe digital recording and store in locked cabinet
12. Code audio recording based on initial coding scheme

All participants were then asked a series of demographic questions (see Appendix B). Next, the participant was taken through a series of questions from the interview protocol which was developed from the theoretical framework which was previously validated based upon the extant literature. There were two sets of questions contained within the interview protocol, one set for the project managers and another set for the team members. If time allowed the project manager was asked both sets of questions. The questions focused on the specific dimensions of team structure, virtualness, and agility. Considering that the theoretical framework and the target audience were well defined no alterations were made to the initial interview protocol.

Due to the semi-structured nature of the interview “probing” for additional information in regard to a specific dimension was a normal occurrence. As mentioned, all of the interviews were digitally recorded and assigned a unique identification number to protect the anonymity of the participant and were stored on a password-protected computer. The digital recordings were then transcribed word-for-word using transcription software into rich text documents to produce a written transcript and imported into MAXQDA for further analysis. Each transcript was given the same identification code of the digital recording for cross-referencing with the digital recording. The transcripts were then coded according to the coding list (see Appendix C) developed from the theoretical framework. In addition to the interviews, documents in the form of team hierarchy charts were collected to indicate the job title, primary role and location of each member.

Overlap Data collection and Analysis, including Field Notes

According to Eisenhardt (1989), "a striking feature of research to build theory from case studies is the frequent overlap of data analysis with data collection" (p. 538). One specific method for tracking this ongoing process of data collection and data analysis is through the use of field notes. This ongoing conversation in the mind of the researcher can be used to record initial impressions, sift through the mounting data, and eventually "these ideas can be cross-case comparisons, hunches about relationships, anecdotes, and informational observations" (Eisenhardt, 1989, p. 539). Pare (2000b) utilized field notes, reflective remarks, and document summaries in their case study project. Running notes, which were entitled "notes to myself" and reflective remarks were utilized throughout this study to keep track of significant details and to provide a mechanism for facilitating the iterative process of theory building research.

Data Analysis

General Analytic Strategy and Specific Analytic Strategy Techniques

According to Eisenhardt (1989) data analysis is the "heart of building theory from case studies, but it is both the most difficult and the least codified part of the process" (p. 539). Similarly, Yin (2003) stated that "the analysis of case study evidence is one of the least developed and most difficult aspects of doing case studies" (p. 109) and that "there are few fixed formulas or cookbook recipes to guide the novice" (p. 110). Therefore, he suggested that case study research should adhere to both a general analytic strategy, to define priorities for what to analyze and why, and a specific analytic technique for detailed analysis of the data and addressing issues of validity and reliability. Ultimately the goal of case analysis is analytical generalization, or generalizing to theory, rather than statistical generalization as in quantitative methods, as such “if two or more cases are shown to support the same theory, replication may be claimed (p. 33).

In terms of a general analytic strategy Yin (2003) suggested three possibilities: relying on theoretical propositions, thinking about rival explanations, and developing a case description. For specific analytic strategy techniques he listed the following: pattern matching, explanation building, time-series analysis, logic models, and cross-case synthesis. For the general analytic strategy this study adopted developing a case description which entailed the development of a “descriptive framework for organizing the case study” (p. 114). Specifically, the theoretical framework was utilized to frame the interview data and to suggest specific propositions in regard to each of the three dimensions: team structure, virtualness, and agility. In terms of the specific analytic technique this study employed within-case and cross-case analysis which will be explained in more detail later in this chapter and will be presented in detail in Chapter 4 (Eisenhardt, 1989; Miles & Huberman, 1994; Yin, 2003).

Data Analysis Procedures

The procedures for data analysis were constructed to identify emerging themes and patterns in the interview data and to generate a logical chain of evidence (Eisenhardt, 1989; Miles & Huberman, 1994; Yin, 2003). MAXQDA, a qualitative data analysis software package, supported the data analysis of the study and was used for multiple purposes. First, it allowed for the coding and categorizing of transcripts.

Second, it allowed for pulling together all data in regard to a specific code from the coding list. Third, it allowed for searching the transcripts for keywords. Fourth, it served as part of the case study database by allowing all transcripts, demographic information, memos, and the research log to be digitally stored in a central location for easy access and retrieval.

The first step was to transcribe the audio-recordings of each individual interview into a text format for interpretation during analysis (Miles & Huberman, 1994). After each audio-recorded interview was transcribed into a word-for-word rich text document it was imported into MAXQDA creating a written transcript for each interview. Each transcript was assigned a unique identification number to protect the identity of the participant and to serve as a reference within the text. The transcripts were then coded with one or more codes based upon the coding list which was derived from the theoretical framework. It was possible that segments of the transcripts were assigned multiple codes. After the coding was completed MAXQDA allowed all coded segments for a specific code to be displayed in a central view. This made it possible to analyze the data for emerging themes and patterns and similarities and differences in the responses of the team members in regard to each question from the interview protocol.

Coding

According to Pare (200b), “codes serve as retrieval and organizing devices that allow the rapid retrieval and clustering of all segments related to a particular question, concept, or them” (p. 12). Miles and Huberman (1994) suggested that one method for creating codes was to start with a provisional list before data collection began. This initial list could be developed from the conceptual framework, research questions, hypotheses, or other key problem areas or variables that were deemed important. This represented a deductive strategy. Consistent with the theoretical framework the coding scheme for this study was created based on the three dimensions of team structure, virtualness, and agility. Codes were also created to capture the challenges represented in the theoretical framework as well. Slight alterations were made to the initial set of codes to better reflect the data as it was analyzed, which is not unexpected (Miles & Huberman, 1994), and the final coding lists is presented in Appendix C. These codes were created in MAXQDA and applied to the interview transcripts according to the questions asked from the interview protocol. A screenshot of the code hierarchy in MAXQDA is demonstrated in Figure 2.

Within-case Analysis

Within-case analysis typically represents purely descriptive, detailed write-ups for each case. According to Miles and Huberman (1994) the purpose of within-case analysis is to draw conclusions based on data obtained from a single case which contributes to the exploration of a “new area and to build or ‘emerge’ a theory about it” (p. 90). Eisenhardt (1989) asserted that “this process allows the unique patterns of each case to emerge before investigators push to generalize patterns across cases” (p. 540). Further, within-case analysis “gives investigators a rich familiarity with each case which, in turn, accelerates cross-case comparison” (p. 540).

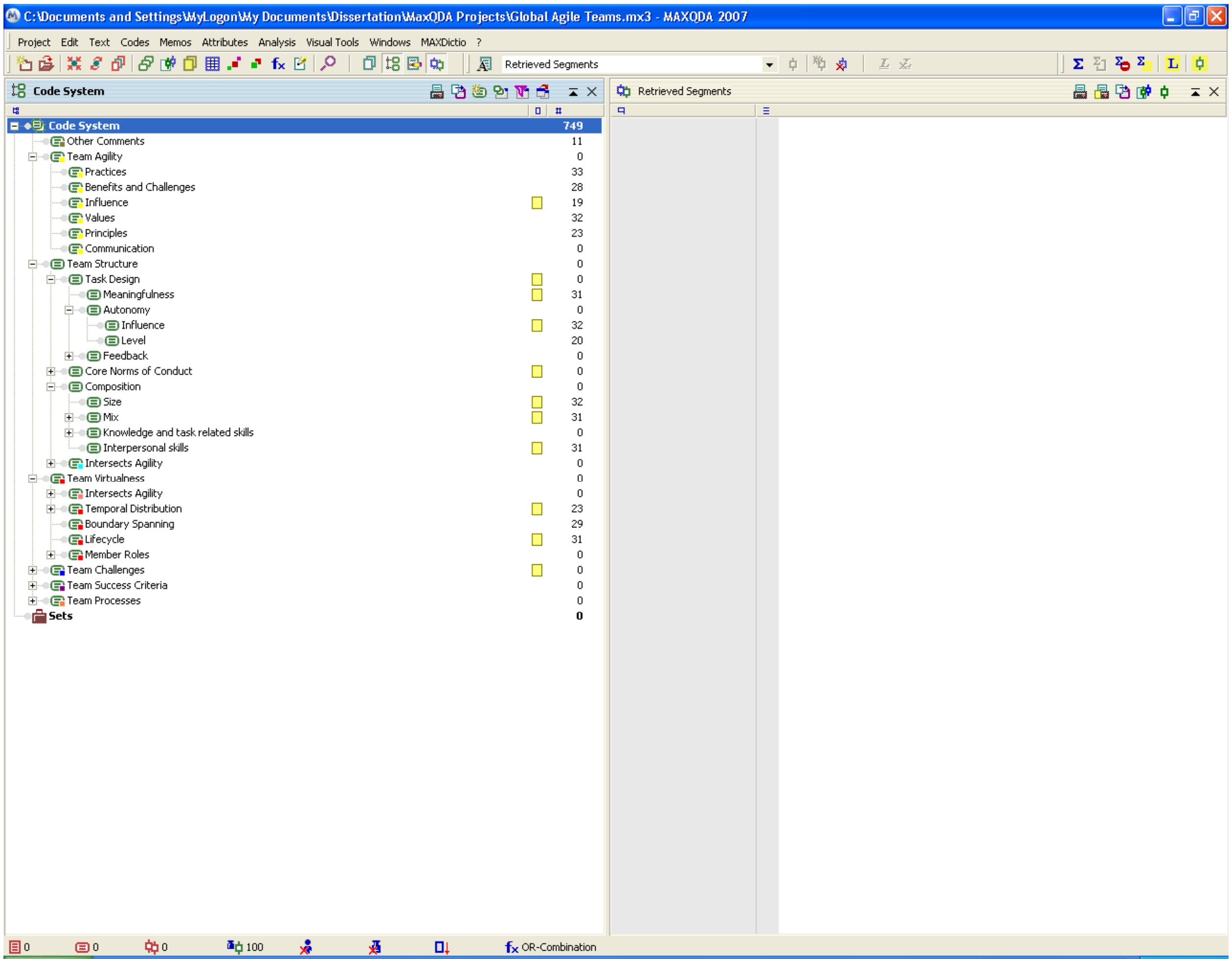

Figure 2. Screenshot of code hierarchy in MAXQDA.

Within-case analysis through the use of these detailed write-ups is important because they are “central to the generation of insight because they help researchers to cope early in the analysis process with the often enormous volume of data (Eisenhardt, 1989, p. 540). With reference to within-case analysis, the “overall idea is to become intimately familiar with each case as a stand-alone entity” (Eisenhardt, p. 540). For this study the cases were analyzed first by sorting text segments by their respective codes to uncover patterns of responses for each code in the coding list. The data were then further analyzed to explore similarities and differences in the cases. All within-case analysis was completed before the cross-case analysis was conducted (Miles & Huberman, 1994; Yin, 2003). The detailed within-case analysis is presented in Chapter 4.

Logical Chain of Evidence

The primary purpose for maintaining a chain of evidence is to increase the reliability of the information in the case study. By developing this logical chain of evidence, the reader of the case study can follow progression from the initial research question to the final conclusions of the case study. This serves to strengthen the construct validity and thus the quality of the case study (Yin, 2003). A primary strategy for linking the chain of evidence is to provide sufficient citations from the case study database in the case study report and developing a case study protocol which indicates the “link between the content of the protocol and the initial study questions” (Yin, 2003, p. 105).

Throughout the within-case analysis comments and direct quotations were referenced with the identification number of individual team members and various matrices were utilized to summarize overall themes for each dimension of the theoretical framework. By using these references and linking concepts to specific individuals and teams this procedure helped to establish a logical chain of evidence by allowing the reader to “move from one part of the case study process to another with clear cross-referencing to methodological procedure to resulting evidence” therefore “increasing the reliability of the information in a case study” (Yin, 2003, p. 105).

Case Study Database

Yin (2003) also recommended the use of a case study database to store the data collected during within-case analysis and the development of a logical chain of evidence using the case study database and the case study protocol. The case study database should be separated into the data itself, and the report of the investigator. The database may include such items as case study notes, case study documents, tabular materials, and/or narratives. In a study conducted by Pare (2002b), the case study database was used to store the following items: 1) raw material, 2) coded data, 3) coding scheme, 4) memos and other analytic material, 5) data displays and matrices, 6) document summary forms, and 7) general chronological log of data collection. This study utilized MAXQDA to create a case study database containing the interview transcripts, demographic information, memos, codes, notes, and research log. Although no standard format exists for conducting such analysis, Pare (2002b) provided a helpful procedure for outlining the steps for the development of a case study database and development of a logical chain of evidence. This procedure was modified for use in this study and is shown below in Table 6.

Cross-case Pattern Searching Using Divergent Techniques

The purpose of cross-case analysis is to go deeper, the "aim is to see processes and outcomes across many cases, to understand how they are qualified by local conditions, and thus to develop more sophisticated descriptions and more powerful explanations" (Miles & Huberman, 1994, p. 172). "Overall, the idea behind these cross-case searching tactics is to force investigators to go beyond the initial impressions, especially through the use of structured and diverse lenses on the data" (Eisenhardt, 1989, p. 541).

Table 6

Within-Case Analysis Procedure

Step		Description
1. Development of a case study database	1.1	Gather reflective remarks and observation notes
	1.2	Codify and extract data from the transcripts
	1.3	Group extracted segments under categories
2. Development of a logical chain of evidence	2.1	Evaluate the team structure dimension of the a globally distributed agile team
	2.2	Evaluate the team virtualness dimension of the a globally distributed agile team
	2.3	Evaluate the team agility dimension of the a globally distributed agile team
	2.4	Evaluate the impact of the challenges to configuring a globally distributed agile team
	2.5	Establish a logical chain of evidence between team structure, virtualness, agility and the challenges to the successful configuration of the globally distributed agile team

Note: Adapted from Pare (2002b).

As such there are two primary reasons to conduct cross-case analysis: (1) to enhance generalizability and (2) to deepen understanding and explanation (Miles & Huberman, 1994). Ultimately, “cross-case searching tactics enhance the probability that the investigators will capture the novel findings which may exist in the data” (Eisenhardt, 1989, p. 541). Eisenhardt stressed that the “key” to effective cross-case analysis involved examining the data in many different ways such as the following:

> 1) select categories or dimensions and then look for within-group similarities coupled with intergroup differences
>
> 2) select pairs of cases and then list the similarities and differences between pairs
>
> 3) divide the data by data source (p. 540-541)

This study employed a case-oriented strategy based upon replication logic (Yin, 2003). The basic idea was to conduct an in-depth study of one case and then to proceed to successive cases to see whether patterns or themes found match those in previous cases until a general theory of globally distributed agile team configuration emerged (Miles & Huberman, 1994). According to Yin (2003), “the analysis can start to probe whether different groups of cases appear to show some similarity and deserve to be considered instances of the same ‘type’ of general case” (p. 135). A strategy for approaching the cross-case analysis involved the creation of word tables to display the common data elements from each individual case “according to some uniform framework” (Yin, 2003, p. 134). In this study the theoretical model provided such a framework. The detailed cross-case analysis and proposition development is presented in Chapter 4.

Matrices for Organizing and Displaying Data

Miles and Huberman (1994) suggested organizing and displaying results from textual data in a matrix form, commenting “tables are for communication, not data storage” (p. 100). Display was defined as a “visual format that presents information systematically, so the user can draw valid conclusions and take needed action” (p. 91). Multiple matrices were utilized throughout the within-case and cross-case analysis to organize and display results from the interviews and to address the research question regarding how agile teams can be successfully configured in a globally distributed environment.

Although these matrices followed the general format suggested by Miles and Huberman (1994) for partially-ordered, conceptually-ordered, and case-ordered displays, the authors themselves indicated that the names were not so important as “how any given display works and how it can further your analysis”; therefore, “all of the displays can be adapted to meet the needs of any particular study; if they seem too complex, simplify. Take what is useful” (p. 101). This “take what is useful” philosophy was applied to the matrices presented in the study. Conceptually-ordered matrices were found to be especially helpful because the combination of rows and columns “bring together items that ‘belong’ together’” and can be derived from a priori constructs from the theoretical framework which it what was done in this study (p. 127). When possible a maximum of one page was devoted to each dimension (sub-dimension) of the theoretical framework (i.e., team structure, virtualness, agility) to reduce the data and to increase the ease of viewing pertinent themes and patterns.

Shaping research propositions

From the within-case and cross-case analysis "overall impressions, tentative themes, concepts, and possibly even relationships between variables begin to emerge" (Eisenhardt, 1989, p. 541). The basic research question of this study addressed how agile teams can be successfully configured in globally distributed environments. In order to approach this question, the theoretical framework consisting of the three dimensions of team structure, virtualness, and agility and challenges to configuration was developed based on extant literature. According to Eisenhardt, in theory-building case research "the central idea is that researchers constantly compare theory and data – iterating toward a theory which closely fits the data" (p. 541). Propositions may define an initial set of laws characterizing the interaction and relationships between the three dimensions of globally distributed agile team configuration and the challenges associated with them (Pare 2002b). The objective was to identify those aspects of each dimension which occurred in successful configurations and attempt to generalize those aspects for the successful configuration of other globally distributed agile teams. For this study propositions are presented throughout the cross-case analysis for each of the three dimensions of the theoretical framework.

Iterative Tabulation of Evidence for Each Construct

The fit between theory and data is crucial to good theory building due to the fact that it "takes advantage of the new insights from the data and yields an empirically valid theory" (Eisenhardt, 1989, p. 541). Eisenhardt suggested two steps for sharpening propositions in order to strengthen this empirically validity: 1) refining the definition of the construct, and 2) building evidence which measures the construct in each case. This is a highly iterative process of constantly comparing the data and constructs, which in this particular study refers to the three dimensions of the theoretical framework. The result is the establishment of construct validity. To address step 1, a discussion of certain constructs is presented in Chapter 5. For step 2, the within-case and cross-case analysis presented in Chapter 4 provide evidence from each team in regard to the constructs of the theoretical framework.

Replication, not sampling, logic across cases

In regard to step two, replication logic was utilized to confirm the emerging relationships between the constructs and thus increase confidence in the validity of the relationships. In situations where a case disconfirms the relationships between constructs, extension and refinement of the theory can be conducted. In sum, "shaping hypotheses in theory-building research involves measuring constructs and verifying relationships" (Eisenhardt, p. 543).

Validity and Reliability of Methodology

In relation to validity and reliability, Yin (2003) suggested that case research should address the four tests common to all social science methods. The first test, construct validity, calls for correctly establishing operational measures for the concepts being studied. This study utilized extant literature, multiple sources of data, and developed a logical chain of evidence to establish construct validity. The second test, internal validity, establishes casual relationships and "is only a concern for causal (or explanatory) case studies" and thus was not addressed in this study. The third test, external validity, establishes the domain to which a study's findings can be generalized. This study employed replication logic and a multiple-case design to strengthen external validity. The fourth test, reliability, demonstrates that the operations of a study can be repeated with the same results (p. 34). Reliability was established through the use of the case study protocol, interview protocol, and the case study database.

Search evidence for "why" behind relationships

In line with Yin (2003), Eisenhardt (1989) argued that qualitative data is "particularly useful for understanding why or why not emergent relationships hold" (p. 542). By discovering the underlying theoretical explanations for the relationships, the internal validity of the findings was strengthened. Pare (2002b) has developed the following tactics for strengthening design quality in case study research which are shown in Table 7.

Enfolding Literature through Comparison with Conflicting and Similar Literature

According to Eisenhardt (1989), "an essential feature of theory building is comparison of emergent concepts, theory, or hypotheses with the extant literature" (p. 544). This process seeks to identify the similarities, differences, and the reasons why. Eisenhardt suggested two reasons why it is important to identify conflicting literature:

1) if researchers ignore conflicting findings, the confidence in the findings is reduced, challenging both internal validity and generalizability

2) conflicting literature represents an opportunity, i.e., forcing researchers

into a more creative, framebreaking mode of thinking than they might otherwise be able to achieve (p. 544)

In sum, “the results can be deeper insight into both the emergent theory and the conflicting literature as well as sharpening of the limits of generalizability of the focal research” (p. 544).

Table 7

Case Study Tactics Adopted for Design Quality

Criterion	Description	Adopted tactics
Construct validity	Establishing correct operational measures for the concepts being studied	Validation of dimensions based upon prior literature and interviews
External validity	Establishing the domain within which a study’s findings can be generalized	Analytic generalization Tying propositions to existing literature
Reliability	Demonstrating that the operations of a study can be repeated, with the same results	Case study database Case study protocol

Note: Adapted from Pare (2002b).

It is equally important to examine similar literature. This helps to tie the similarities together and associate the phenomena. Thus, internal validity is strengthened, generalizability is widened, and the conceptual level is increased (Eisenhardt, 1989). This linkage among the literature is “particularly crucial in theory-building because the findings often rest on a very limited number of cases (Eisenhardt, 1989, p. 545). Chapter 5 presents a discussion of findings from current literature studying the use of agile methods in globally distributed environment and provides a comparison of conflicting and similar literature.

Reaching Closure through Theoretical Saturation When Possible

Eisenhardt (1989) suggested two important issues in regard to closure. First, when to stop adding cases, and second, when to stop iterating between theory and data. Ideally, “researchers should stop adding cases when theoretical saturation is reached” (Eisenhardt, p. 545). Theoretical saturation refers to the fact that “no new or relevant data seem to emerge regarding a category, the category is well developed in terms of its properties and dimensions demonstrating variation, and the relationships among the categories are well-established and validated” (Strauss & Corbin, 1998, p. 212). However in regard to practicality, Eisenhardt (1989) stated that it is “not uncommon for researchers to plan the number of cases in advance” (p. 545).

Eisenhardt (1989) suggested that between 4 and 10 cases is adequate. Having more than 10 cases significantly increases the complexity and the volume of data that must be analyzed. In sum the decision to stop adding cases and iterating between theory and data is “when the incremental improvement is minimal” (Eisenhardt, p. 545). In this study the determination of adding cases was be based on the need to adequately address the dimensions of the theoretical framework. Another determining factor was the number of teams available within each of the organization for participating in the study. Across the five teams interviewed it became evident that information was starting to be repeated indicating that theoretical saturation was being reached.

CHAPTER 4

DATA ANALYSIS

This chapter presents a brief description of each organization and associated team which participated in the study. The descriptions are limited in an attempt to suppress any identifiable information on the organization, the team, or the individual members. The within-case analysis and cross-case analysis are also presented with associated matrices summarizing the findings from each team.

Organization A Cases

Organization A Description

Organization A is a global IT services company based in the United States providing IT solutions to customers around the world. It employs a diverse group of approximately 135,000 employees world-wide. It addition to company employees it utilizes contract employees from various locations around the world. As a whole the organization still follows a structured, waterfall approach to software development as a part of its standard procedure. However, small pockets of individuals and teams are beginning to slowly implement the values, principles, and practices associated with agile methodologies. Currently this is being done primarily by a “champion” or “advocate”, someone who is trained in agile methods and is leading the effort.

Team A01 Description

The following section provides basic information about Team A01 including: size, how members were selected, locations, type of project, how long the team has been in existence, how long the team will be together, familiarity with the Agile Manifesto, which agile methodology and practices have been implemented, what type of agile training the team has received, other projects the team has worked together on, and the success rating of the team. Table 8 summarizes size, selection, locations, and project of team description:

Table 8

A01 Team Size, Selection, Locations, Project, and Duration

Size	Selection	Locations	Project	Duration
37	Background, skill set	Ten cities across seven states in the United States; One city in Australia; One city in India; One city in Mexico	New software development in a transportation-related industry	Team has been in existence for approximately one year and will be a long-term team

The team has been in existence for approximately on year. Members have moved on and off the team in that time. The team began with the project manager who then began to slowly add project leads, technical leads, and the team was ramped up over the course of the year to the its form at the time of this study. The ramping up took a little bit of time to get the new project started and to establish the scope. Seventeen members participated in the study. This particular team will eventually evolve into a production support team. At the time of the study the team was involved in building a brand new product which did not currently exist. The development will span approximately three years. At completion the product will begin serving clients and will transition into a production support mode and will then provide ongoing enhancements for the product. Table 9 summarizes demographic information about each member who participated in the study while Table 10 summarizes the title, location, time zone, status on the team, and study participation.

The team was not following a specific agile methodology in its entirety. In general the team implemented a modified version of Scrum. As a part of the implementation of a modified form of Scrum the team has adopted specific agile practices including the daily Scrum (daily stand-up meeting), Sprints (short iterations of six weeks), and time-boxing. The team has also implemented additional practices that are commonly identified as agile including: Test-Driven Development, iteration planning, velocity/estimation, and small releases.

According to one member on the team the recognition of agile methods was more predominate at the project management levels. Thus the project leads were familiar with agile, while the programmers probably were not, “their task are a little bit . . . they are a little bit more removed from that, so I would say the leads yes, but the programmers probably no” (A01001). In terms of familiarity with the Agile Manifesto it was at a very high level. Perhaps a few members knew quite a bit about it, but in general everyone was in the process of learning more about agile, and “I think the vast majority do not” (A01001).

Table 9

A01 Demographics for Members Who Participated in the Study

Title	Location	Employment Duration (in years)	Present Position Duration (in years)	Team Member Duration (in years)	Allocation
Programme Manager	USCity7	17-24	9-16	1-2	Main
Project Management SME	USCity5	> 25	3-4	< 1	Part
Consulting Architect	USCity10	17-24	9-16	< 1	Part
Consultant Senior	USCity4	5-8 yrs	1-2	< 1	Part
Communication Infrastructure Specialist	USCity1	9-16	9-16	< 1	Part
Senior Information Specialist	AustraliaCity1	17-24 9-16		< 1	Part
Information Specialist	USCity8	9-16	< 1	1-2	Main
Project Leader	IndiaCity1	< 1	<1	> 3	Main

(table continues)

Table 9 *(table continued)*.

Title	Location	Employment Duration (in years)	Present Position Duration (in years)	Team Member Duration (in years)	Allocation
System Consultant	USCity9	9-16	9-16	< 1	Part
Project Manager/SME	USCity6	5-8	5-8	NA	Part
Business Analyst	USCity9	< 1	< 1	< 1	Main
Lead Architect	USCity6	5-8	< 1	< 1	Part
Business Development Lead	USCity8	17-24	< 1	< 1	Main
Architect USCity4		17-24	5-8	1-2	Main
Senior Analyst	USCity6	< 1	< 1	< 1	Main
Consulting Senior	USCity4	1-2	5-8	< 1	Main
Project Office Leader	USCity6	9-16	1-2	1-2	Part

Table 10

A01 Current Members and Geographic Locations

Title	Location	Time Zone	Current	Study
Programme Manager	USCity7	GMT -06:00 / CST	X	X
Project Manager	USCity6	GMT -06:00 / CST	X	X
Project Management SME	USCity6	GMT -06:00 / CST	X	X
Architect/Technical Lead SOA	USCity3	GMT -07:00 / MST	X	X
Technical Lead	USCity8	GMT -06:00 / CST	X	X
Project Management SME	USCity5	GMT -05:00 / EST	X	X
Architect / Consulting Senior	USCity4	GMT -06:00 / CST	X	X
Analyst / Consulting Architect	USCity10	GMT -06:00 / CST	X	X
Project Office Leader	USCity2	GMT -06:00 / CST	X	X
SOA and Lead Architect	USCity6	GMT -06:00 / CST	X	X
Business Development Lead	USCity8	GMT -06:00 / CST	X	X
Business Analyst	USCity9	GMT -05:00 / EST	X	X
SOA Analyst/Architect	USCity10	GMT -06:00 / CST	X	X
SME / System Consultant	USCity9	GMT -05:00 / EST	X	X
Project Leader	IndiaCity1 GMT	+5:30	X	X
Com. Infrastructure Specialist	USCity1	GMT -08:00 / PST	X	X
Project Manager & Lead	AustraliaCity1	GMT +10:00	X	X

(table continues)

Table 10 *(continued).*

Title	Location	Time Zone	Current	Study
Project Manager	USCity6	GMT -06:00 / CST	X	
Architect/Area Lead	USCity9	GMT -05:00 / EST	X	
Business Analyst	USCity9	GMT -05:00 / EST	X	
SME	USCity8	GMT -06:00 / CST	X	
SOA Director	MexicoCity1 GMT	-06:00	X	
SOA	MexicoCity1	GMT -06:00	X	
SOA MexicoCity	1	GMT -06:00	X	
SMA/XML	USCity1	GMT -08:00 / PST	X	
Unspecified	USCity1	GMT -08:00 / PST	X	
Unspecified	USCity1	GMT -08:00 / PST	X	
Com. Infrastructure Specialist	USCity1	GMT -08:00 / PST	X	
Unspecified IndiaCit	y1 GMT	+05:30	X	
Technician	USCity9	GMT -05:00 / EST	X	
Technician	USCity9	GMT -05:00 / EST	X	
Technician	UsCity9	GMT -05:00 / EST	X	
Quality Center Admin	USCity9	GMT -05:00 / EST	X	
Programmer IndiaCity1		GMT +5:30	X	
Programmer IndiaCity1		GMT +5:30	X	

(table continues)

Table 10 *(continued).*

Title	Location	Time Zone	Current	Study
Programmer	IndiaCity1	GMT +5:30	X	
Programmer	IndiaCity1	GMT +5:30	X	

Note: GMT = Greenwich Mean Time; CST = Central Standard Time; EST = Eastern Standard Time; PST = Pacific Standard Time; MST = Mountain Standard Time.

Training was provided as aspects of agile were needed. The training involved primarily the project leads. At the programmer levels training was focused on the actual use of technical tools or specific programming languages. Due to the fact that this team consisted of full-time and part-time members, some people were also working on other projects in development, production support, etc. As a group this team had not worked on any other projects, agile or otherwise.

Rating of Team Success

In terms of success, one member responded, “so far, yes” (A01001). Some of that success to-date was related to the ability to work across time zones and conduct the daily Scrum to discuss technical issues. In terms of increasing the success of the team this member indicated that the use of more full-time people would be beneficial:

> There are sometimes when I wish I could get fulltime people instead of part-time people, so we have a lot of, probably half the team is part-time, so, as far as successful team to be more successful, I wish that some areas had more fulltime

> dedication, because sometime with part-time you get pulled people off the project to work on higher priority things, whereas if they had been dedicated fulltime to the project that probably would not have happened (A01001).

A01 Within-Case Analysis

The following section provides a description of the team and analyzes the interview data to explore the response of members in regard to the three dimensions of the theoretical framework: structure, virtualness, and agility. The analysis of each sub-dimension concludes with a summary of the potentially significant findings.

Team Structure

Task Design

Meaningfulness. As one member pointed out, the team's feeling about the meaningfulness of the project was important to the level of effort put forth by the team, "I think if they have that feeling it is incentive to work harder" (A01009). One of the main themes emerging from Team A01 in relation to meaningfulness was the importance of the visibility or criticality of the project as indicated by the following member, "it assumes all the more greater importance when it is a critical project or it has high visibility" (A01008). Another member stated the importance of visibility this way:

> Absolutely, I think the more visible, the more dedication you would be . . . above and beyond what 's expected, meaning an individual might put in their normal hours on any project and delivery on time, but if the project is visible they might put in extra hours on their own times to ensure certain other things (A01005).

So meaningfulness was in a sense tied to the visibility and privilege of working on new technologies and new development, “so I think they know that and it’s involved so many new technologies that it’s a privilege to get to work not only on new development but all new technology so that’s exciting . . .” (A01001). Thus in a situation where there is not a lot of major new development in the works, when a project comes along that is development a new application, that brings a heightened sense of excitement and motivation to be chosen to work on that new project. It also motivates members to work harder in order to stay on that project, “as things go they know that they rotate off a project depending upon how many resources are needed so I think that everyone wants to continue working on that project” (A01001). This emphasis on the visibility or criticality of the project continued in the response by another member of the team:

> But if you know the project is critical or, you know, say for example in some of the projects there is some new technology being factored. . . that you get exposure for the first time . . . there is the possibility that team members, you know, get more involved, give their more effort than a just . . . another project which is just making changes to an existing thing (A01004).

In a situation where the organization has made a substantial investment into a product and a person is ask to be part of the team for that project, “. . .the individuals are definitely are more motivated because they realize the stakes involved” (A01015).

A second aspect related to meaningfulness was the members overall understanding of the project, “. . . but I think any time they understand the project well I think they are going to work harder and better” (A01002). A part of that understanding encompasses knowledge of the scope of the project and what it involves:

> I think it is important for everybody on the team to know the full scope of the project as well, I mean, that is another aspect of it, cause I think it only helps the team success if everybody is aware of what the project entails and that . . . for that side of their team but, you know, another team involved in the project, all necessary as well, so I think the success of the project also requires everybody to know the full scope, not just maybe the area that they are involved, but really what is going on in the whole project (A01006)

This provides the team with the big picture and allows them to be aware of what other members are involved with in the project.

Experience level was also identified in relation to meaningfulness. It was suggested that new members to the team or industry or those with less experience are more inclined to put forth greater effort “when they are actually new to the industry and the domain and the whole IT development environment that tend to put in more effort than is normally required” (A01008). The argument for this perspective is stated by the following member:

> I think it would be more true of people with lesser experience. I think maybe people with more experience maybe start taking, they . . . just cause they are more professional and have more knowledge and so on. They can work, you know, in whatever situation it is . . . (A01006).

This perspective deemphasized the importance of the project itself and placed more priority on the person and their experience with the organization:

> I would say that it does not depend on the project, it depends on the person and how long they have been in the organization. I would say . . . well . . . for me

> personally, yes, because I know what it is I am building and so I would be more personally invested. But I can't really speak for all of the team members (A01011).

This idea of personal interest was echoed by another member of the team, "I think if they're allowed to personalize it before they can, you know, provide value to the organization. So, if you are personally not interested in something, you are not going to provide value to the organization" (A01014). An individual's role on the team and their level of responsibility might also impact the link between meaningfulness and motivation in that project management might be more vested in the project, it is more with the, you know, team lead and other top people, the team I think is more of a driving force for them (A01016). A final aspect related to the personal, individual element of the team was the "pressure that others are depending on you to get finished" (A01001):

> They know we have some deadline that have to be met and they know that there work affects other people who are working on that, so I think that knowing that they're one link in the chain and that other people cannot get finished till their work is done probably puts more pressure on them than if they were on an independent project – so in that regard umm there is pressure to get the work done faster as you know that others are waiting on you (A01001).

Creating a product which was beneficial to the customer and having a sense that the project has the potential to be successful were also cited as important aspects related to meaningfulness and motivation, "if they feel that they're developing a product . . . they are in a project that is going to benefit the customer I think they all, they are working hard and are motivated to produce excellent results" (A01013). Similarly, another member commented, “when you know what you are working for and you know something is tangible, meaningful with some productive output, yes, people work much more, much harder” (A01012). Members want to feel that the project can succeed, that they can produce a beneficial product for the customer. If there is a sense that the project does not have the support or resources to succeed that can decrease its meaningfulness and thus lessen motivation:

> I mean it also depends on the probability that they are going to succeed. You could have a really you know socially important, you know, project, but if you perceive the . . . if your expectation is that the project isn’t going to succeed I do not know how motivated people are going to be (A01003).

One member did point out the fact that the team must be careful not to place an inordinate amount of effort on what they deem more meaningful, or more important. As indicated by the responses above, meaningfulness does appear to be significant in the level of motivation and effort, but it is also important that team members respond professionally regardless of the project to which they are assigned:

> Well, I don’t think they make that judgment. A project is a project is a project, and you know, it is always presented as good to do for the client, whether it is internally funded or client externally client-funded, I don’t think the developers or the team at large would make any judgment whatsoever as to, oh this one’s more . . . my project is more important than yours, so I am going to work harder (A01010).

The following section provides a description of the team and analyzes the interview data to explore the response of members in regard to the three dimensions of the theoretical framework: structure, virtualness, and agility. The analysis of each dimension concludes with a summary of the potentially significant findings.

Autonomy. When considering autonomy within a team and its control structure it is important that it be designed to facilitate a collaborative environment, "I think that when team is setup to foster communication and openness, each person having a voice, I think your project is a lot more successful than if someone is not fostering that type of mentality" (A01013). The term autonomy carries with it the idea of self-governance, of having some control over what is done, when it is done, and how it is done. It carries with it some flexibility in the decision-making process for each individual team member to some degree. The overall feeling within this team was that establishing clear responsibilities, assigning capable leadership, and defining plain processes can have an influence on the impact of autonomy on the team, whether positive or negative. Because of the distributed nature of this project, it was important to first decide who was to oversee a particular area:

> . . . as you know our project is distributed, rather the resources on the project are very much distributed, in fact I don't think there are more than a handful of people, more than two or three people in one location throughout our entire project, so what that also means is that we have to divide up lead responsibilities and I think that it is, it's solidified, as we first got started I think it was hard to tell who was over what areas, I think it kind of impacted the speed at which things got done (A01007).

In a distributed environment, “people who manage the resources at every location they should be given the chance to manage and plan individually and that would work best for the overall organization and the team, that is my opinion” (A01008). From this member’s experience, “I think autonomy does help catering to the team’s aspiration better than, I know, remote controlling a team from a distant location” (A01008).

So much of the success of the project depends on the leadership of the team. It is crucial that a sense of openness is conveyed and that members are free to ask questions and make decisions:

> I think it has an influence and I have seen it go both ways. I think it really depends on the leadership of the project and the stage that’s set for everybody’s got good ideas, where all in this together, no question is a bad question and that is really the environment we try to work in, although, we’re not always 100% successful (A01010).

It is the leadership that sets the tone for the autonomy of the members.

> The technical leadership and the project management or program management goes a long to setting that feeling of autonomy and I can make decisions and the empowerment, give me the general guidance and direction and I will get you there and that kind of thing (A01010).

However, the leadership must also recognize that in some situations where a major decision must be made that the institution of proper process must be put into place and that in these situations more rigidity may be necessary:

> . . . we have certain processes we are expected to follow and we have certain reports we are expected to produce and those definitely go up the chain and if

> there is a major decision to be made about how to go forward with something then it has to go through that chain, it is very rigid in that respect (A01015).

The experience level of the leadership with the project and with the customer may also have an influence on the attitude of the members. If a project manager or technical lead is constantly changing their minds in the middle of the project or wants to go a different direction on a regular basis,

> . . . then it kind of generates some hesitance as far as getting something done and delivered versus if you've got the experience with working with that particular group or client . . . what they want and then you know what you are doing is not going to change in such a short time (A01005).

Not knowing the direction of the project or what is expected may cause the members to lose trust in the leadership and not want to assert too much autonomy for fear that their work will be of no value when the leader changes their mind or does not have a clear expectation of the customer's requirements. In sum, the leadership can definitely have a positive influence on the team in a globally distributed environment:

> Yeah, I think it is positive. I think it's good to, you know, positive to have, in our case for example it is called the program manager, could be project manager, but it is someone who is capable, in control, I think it's good to be flexible as well, I think in our situation where we have a lot of people in different locations, globally and within the US, then you also have got to be flexible, that person in control also got to be flexible, but they have to exhibit some control and I think everyone's, I mean, I have been involved in some projects where maybe, you know, the project manager, and so on maybe isn't as controlled as in other

> project, I think over time it will affect the project and you it could bring in inconsistencies and, you know, direction of the project maybe going down the wrong way and all sorts of different things happening. So, yes, I think it is very important that you see somebody who is in good control of the project. (A01006)

Autonomy must also be approached from the stand-point of the project itself in addition to the people working on the project. Some projects may lend themselves to a higher degree of autonomy than others in terms of identifying and partitioning the . . .

> deliverables in such a way that you could complete chunks of work in an autonomous manner. And I think that some projects where you can certainly do that and probably would work just fine and other projects where you couldn't chunk it up in that manner, so it really just depends (A01003).

Additionally, there are some areas of the project which may be considered “critical” and others which are “streamlined” which are the terms ascribed by one of the members of the team. This member defined a critical area as “architecture, as far as leading and establishing some kind of standard. Once we do that pretty much everything else is kind of autonomous” (A01012). Once the standards and processes, the architecture, are established by the project manager or lead, “the testing team can then do there job, the development team can do their development” (A01012). Consequently, it is important to look “closely at those two roles and who is performing it does have an impact on the overall result that you get from this” (A01012).

When managed correctly self-organization and shared decision-making can have a positive influence on the attitude of members “if you know that you are autonomous and you can, you know, you can organize your work however you want as long as you meet your deliverable. I think that does help a lot. Not being micromanaged does help" (A01011). Subsequently, if project management "has too tight of control, then of course the members don't like that . . ." (A01004). In regard to the positive affect of autonomy on the team, one member made the following statement in relation to the role of the developers and designers:

> It has a positive, I think. What we have tried to do with the developer is that, you know, we have told them that they, you know, the have the ability if they see something that is wrong or needs to be change, then they can change it, so that they feel like they some control and not just blindly doing whatever was written down. As far as the designers, they feel in control and so therefore they have a positive impact (A0104).

Ultimately in terms of autonomy the following comment captures the essence of how it should be approached in regard to the members:

> I think they need to have enough freedom, they need two things, they need one to understand how their tasks fit into the big picture, so they know how their actions affect others, and then they need to be given the freedom to do the work assigned to them, without being micromanaged. But, the flip side of that is that they do have to understand the critical path of the project and how their work affects others (A01002).

In sum as one member commented, balanced autonomy "I think it has a positive impact on the success of the team" (A01016).

As indicated by the responses of the team members autonomy appears have a significant influence on the configuration of the team in regard to task design. Members also indicated the level of autonomy that thought was necessary within a globally distributed agile team on a continuum of low, moderate (A01005; A01008; A01013), or high. Overall the consensus showed that members suggested a moderate to high level of autonomy.

Those members suggesting a moderate level of autonomy provided the following reasons. One reason identified was the virtual or distributed nature of the team. One member provided the following rationale for choosing moderate over high:

> Okay, I only have three choices (laughter). So it is going to sound funny I am going to go with moderate, and the reason I say that is that my experience with A is that we are very, very virtual. So, if I had a colocated team I might say high, but seeing as how we are very, very virtual, I think that we need . . . there are just so different many pieces and that we have to make sure everything is flowing together right, so I want them to have as much autonomy as possible without losing sight of the pieces fitting together. That is why I went with moderate versus high. (A01002)

In a globally distributed environment it can be difficult to keep track of the team member's progress and to coordinate the project so that everyone is on the same page and not developing in their own direction:

> Well, the further you are spread out, the harder it is to keep track of everybody, so I would say moderate because you need to have some sort of control because if not you each will be developing in their own directions and might not met up at the end. For globally distributed teams moderate, I mean, if you're located together and I guess it could be higher autonomy, the further apart you are, I

think the more you have to, you know, show what it is you did all day (A01011).

A second reason revolved around keeping the lead in the loop without the team being micromanaged. There should be a balance between control of the project management team and the other members of the team:

> I would go with moderate, to the extent that, you know, that whoever is controlling, like the lead of the team, should be kept in the loop of what everyone is doing, but then he, he or she should not exercise too much control and it shouldn't be like, you know, micromanagement type of thing. But then still the team members should have, you know, full autonomy on what they are working, but then still it should be little bit of control on the team members, shouldn't happen that, you know, the lead doesn't even know what the other team members are doing (A01004).

This balance of control was echoed by another member who stated, "for a team to be successful, a high level of autonomy . . . no, I think medium. I think they should have some autonomy, but they do need to be controlled" (A01009). Again, some degree of control is necessary within the team so that the product meets the requirements of the customer while at the same being programmatically feasible:

> I would say moderate. You need to trust people to make decisions on their own when it comes to creativity and obviously, you don't want a programmer out there making decisions about this requirements doesn't work real well with this way programmatically so I am going to change it. There has to be some level of control so that you're going to come up with a product that you anticipated getting

> that was ask for, but other that, in their day to day decisions I think people should be allowed to make . . . to have some autonomy (A01015).

Those members who recommended a high level of autonomy also specifically cited the global distribution of the team as a primary reason, “right, I think that especially since it’s globally . . . it’s global that you may not have someone next to you, you have to be able to have the ability to make decisions, so, yeah, I think high, definitely is necessary” (A01014). This sentiment was echoed by another member on the team who stated the following in regard to time zone differences:

> I think it needs to be high because if somebody’s sitting a time zone or two, or three away needs to make a decision to get on with it, they cannot necessarily afford this 24 hour and that’s what it turns in to. Somebody sends a question at the end of their day and we get it at the beginning of our day and send it at the end of our day and then they get it at the beginning of their day, I think the autonomy state needs to be really set high (A01010).

Due to the fact that the team is not colocated there is a need for a reasonably high level of autonomy as well as flexibility:

> I, personally, you know, think it is going to be reasonable high, but it’s got to be flexible like I said though, if you don’t have anybody, everybody in the same room and/or same location then you really got to be flexible. That is important, but I think it should be in most projects, there should be a reasonable high level. Yeah. You know you don’t want it to be excessively high, somewhere between moderate and high I would actually rate it (A01006).

However, there does need to be a balance so that high autonomy does not cause a negative impact on the team as this team member continued:

> Because if somebody is excessively autonomous I think then that may, a swing, a negative impact on the project where because there is a feeling that somebody is just way too much control and is not flexible and so on. So, I think, you know, moderate to high, but there's got to be a level of flexibility (A01006).

Feedback. The influence of feedback on the configuration is very important, "I think the more feedback they receive the better. The more successful they are going to be" (A01002). This is true whether the feedback is positive or negative as explained by the following member:

> Whether it is positive or negative feedback, especially well positive definitely, but then especially negative because if somebody is not doing it . . . you know, his or her work correctly, it . . . the person knows it ahead of time instead of, you know, after the project is done . . . then it is not time for the feedback. They have a chance to improve upon (A01004).

While explaining the feedback cycle for an internal client, one member indicated that that multiple meetings to gain input are scheduled each week. Meetings with supervisors, client representatives, project mangers, and technical leads occur on a weekly basis. This provides a continuous feedback. In addition daily meetings are held very day for fifteen minutes, "they are very focused on what you did and the issues. So I think it is very important to have high level feedback" (A01001). These daily meetings are conducted via teleconference "where a lot of things addressed and feedback between team members, components, etc." (A01005). This daily meeting is somewhat dependent upon where the team is within project, as this member suggested, a weekly call is sometime sufficient, but “as you get to the end then daily might apply but may not be necessarily beneficial, but it is a short meeting . . . so” (A01005).

Although it is important to get feedback from persons outside of the team, it is equally important to elicit input from individuals on the team because they are the ones actually working on the project:

> I think that’s very critical, I think you need input from the participants because they are the ones who are participating, they are the ones who have an active role and their input is probably more important and reliable than output outside the project (A01006).

Additionally, input from more seasoned members who have been a part of a similar process and may have suggestions about how the project might evolve or give suggestions on how to deal with new situations based on past experience:

> You also need input from some experienced people who have gone through the same process before, I mean I would imagine that every project is different therefore you have to input from your own team members. And in a lot of cases a project evolves . . . anyway as it goes along so . . . you need input to maybe . . . to maybe manage the evolving of the project a little bit (A01006).

So as requirements, personnel, or other changes occur it is critical to have continual input from everyone involved in the project and “to setup a situation where its people feel that their input can always be provided and in . . . it is not shunned or frowned upon” (A01006).

In a globally distributed environment the team found that agile practices can help in gathering feedback from members of the team at the different locations:

> Yeah, I think that is real critical especially when you are not around them so you can't tell just by looking around to see who is needing, needing more work or is waiting on things, you have to stay on top of the little hints, we do have daily Scrums, that is one of the agile methods we have employed, so we do find out on a daily basis who's needing more work or who's waiting on feedback, but you lose time in that, that's, you know, is at four o'clock, the person may have been waiting for some time, or people are not around, you do get the sense that work could be done more efficiently with faster feedback, and feedback opportunities, so and you do get less of that when everyone is remote (A01007).

Thus, having a mechanism for meeting on a daily basis and finding out where members are on the project can facilitate to the feedback loop "and the Scrum accomplishes that . . . helps that a lot" (A01015). According to one member, an unwritten guiding principle for the team is "be upfront, blunt, and candid" (A01010). It is so important for the team to know how things are going and what needs to be improved upon or dealt with, as this member continued, "in other words say what the hell you are thinking cause that's the only way we are going to get to the bottom of it, and, you know, that works, it's tough sometime, but it works pretty well" (A01010).

The source of feedback has two primary sources, either personnel such as upper management and project management or the work itself via working code and application functionality. For some members they prefer feedback coming directly from the work being done, "because that is immediate, so I can see whatever I am doing is actually working on . . .” (A01011). Others, however, though they recognized the benefit from gaining feedback from the work itself indicated that “it seems to work better when the manager or the leader provides feedback and encourages the team and identifies the pain points and helps improves productivity” (A01008). To some degree the source of the feedback is dependent upon the person:

> . . . some people really just want to go task to task to task and get things done and see that what they are doing is working and that is good enough for them. But, others we have had on our team very much want to know that what they are doing is, that it all fits together into a grand plan, and that’s, you know, that it is all going to work in the end (A01007).

The general feeling of the team, however, was that both sources of feedback are necessary, “it is extremely important wherever it comes. But, it should come from both ends I think. I think communications is so important in any project. You know, it should be from both ends” (A01009). Although it is beneficial for members to “see the results of performance metrics or critical path on the schedule. I think it is very reassuring to them and it is very important, but I still think they like to hear from their leader” (A01002). A balance of all the stakeholders involved and the feedback from actual physical progress on the application are both essential to a successful project. In sum, a member of the team commented:

> My initially thought is there needs to be a balance between both. In any . . . in any project you have different stakeholders, I think feedback from all the stakeholders to the success of a project and those stakeholders in this case you are saying are outside influences versus internal and I think at different points in

> the project, the outside influences may have a lot more . . . what's the word . . . emphasis or importance than within the project, but then it switches at other stages. You have to keep all the stakeholders in . . . you have to balance all of the stakeholders (A01013)

Core Norms

In general core norms of conduct are not project specific. Standard organizational policy and procedure influences the team in such common areas as those outlined below:

> Organization A does have mandatory type training that we all take ongoing, different types of things like confidentiality for our clients, when we deal with data, certainly there is code of conduct that we all take training so we all understand again confidentiality, how we operate, so at a high level or general level we all take that regardless of the project we work on (A01001).

In addition there are team standards in place for the project including reporting, checking in code (i.e., not keeping code checked out too long), walk-throughs, and the building of software (A01001). Another member recommended that norms in relationship to escalation path are also important:

> . . . when it is time to escalate something. You don't want them sitting there stewing about it, you want to know how to bring up an issue and who to take it to. And what the proper manner for doing that is. So, I think it is very important. Or else they are going to sit back and stew about (A01002).

However, it was noted, that the organization must be careful not too impose norms to heavily in an effort not too restrict the work of the team (A01014). In regard to this organizational perspective, one member commented:

> We have a common code of conduct that is entertained all over the world and we are very particular about people following the code of conduct and understanding what it is and following it in letter and spirit (A01008).

In sum, with the global nature of the organization, another team member noted the relationship between company norms and the offshore sites, “you know, I think, we try to develop a culture that defines some of those norms, but other than that nothing is really spelled out with the offshore, it is just the company’s standards” (A01016).

Much of the influence of core norms was related back to the overall “corporate culture” (A01010). There were multiple comments about the professionalism within the team which makes the establishment of a strict set of core norms of conduct of less significance, “I think it’s . . . I think it’s becoming less important simply because people are used to dealing with it and again it goes back to experience and professionalism, I think” (A01006). In regard to the relationship between professionalism and core norms this member continued by stating:

> And I think over the years as you get used to it everyone knows what to expect what to do and so on. So, although those things are important to be defined at the start of a project, I think it’s not as formal as it used to be simply because everybody acting professionally and knows what to do, and so on. (A01006)

Most members recognized the importance of core norms, but in terms of having something like the “Ten Commandments” that everyone has to follow, that is not necessarily a requirement. As the following member commented,

> Absolutely, I think it is very important to establish that, but would you specifically, verbally, or, you know, publish the establishment, I am so sure. Most people are professionals, they know exactly what to do and how to handle it . . . (A01012).

Core norms should be left flexible enough so that tabs can be kept and addressed if something “goes out of hand and, you know, establish a process around that to keep things in check . . . leave things implicit, you do not need to make it explicit” (A01012). Echoing this sentiment in regard to a formal written set of norms another member stated, “I don’t even think it is required, because we are all professionals, so there is some, you know, you expect professionalism from each team member, and typically you see that” (A01004). In regard to question about the establishment and influence of core norms on the team, one member responded in the following way:

> Yeah, it’s very important that people are professional and that they work together well. The environment needs to be favorable for that type of work. I guess I don’t know that seems like a silly question, because one of the expectations in a good, functioning team is that everyone is respectful of one another and that behavior is considered, you know (A01015).

Although professionalism was noted as somewhat of a given within the team, one specific recommendation was to establish any particular norms, not necessarily formal or written, during the startup of the project and then trust the members to act accordingly based upon those norms:

> But, you know, at the beginning of the project, the project manager or the program manager should establish those rules, and it may be only a short effort to do that, but yes, they should be established, but I think nowadays everyone acts professionally any way and adheres to all those and you know, a lot of those are unwritten rules and unspoken rules but everybody adheres to them. (A01006)

Working in a globally distributed environment does have an impact on the establishment and influence of core norms on the team, these are “very important when you start thinking of the global aspect of it” (A01005). As the following member stated:

> It is very important to have a successful project, especially in, you know, different areas, the distributed setting. Yes, I think it is very important. Everybody should be on the same page all the time, and doing the same thing. (A01009)

However, as one member noted, sometime the norms are not as evident as they might be in a colocated environment because you are not physically around other members:

> You don’t really even notice, I am inclined to think that it is less so, because you are not around, you know, you are kind of talking to an instant messaging session or you are talking to someone over the phone, it’s not . . . you don’t have the opportunity to really get to know your team as well as if you were in a bullpen somewhere. So, you don’t . . . , you know, the conflict hasn’t . . . I haven’t seen noted any that we have had, I haven’t seen any (A01007).

So from this member’s perspective they have noticed fewer conflicts that need to be addressed because of the distributed environment.

Culture was also associated with the development and influence of norms within a globally distributed agile team. One member related the importance of norms to the work that place in the United Nations:

> I think it is very important. I have for so many years I have worked on development projects in an environment where it was like working in the U.N., we had all different types of cultures and all different types of people working in the same building and understanding their cultural differences and accommodating language barriers and things like that, being tolerate of all that was very, very important to success. In this particular project we are also dealing with that and I think that because we are an open and honest communication project I think that just being tolerant of the different cultures is a good thing (A01013).

Another member stated that with people of different cultures coming in and out of the team that:

> you cannot expect people to be diverse 100% because it comes with exposure, so there is always something that might come up, you know, as you add more people depending what country they are from and their customs, etc. You know, what language they accept or don't, what's important, so it is a trial and error really, a learning process. It is important to have some set rules. (A01005)

A specific example was provided by a member of the team about the issues of cultural, language, and interpretation challenges that are inherit with global teams:

> Now as far as cultural differences go, we have some . . . part of our team is in Mexico and we get used to the two hour lunch, the 2:00p, 2 hour lunch which is kind of interesting, right smack in the middle of the day, and of course the

> language differences are interesting but they have not . . . there have been times where you are really not sure whether something is being understood properly and you have to kind of go over the same ground a couple of times to make sure that things are getting understand. But all in all it has worked out pretty well (A01007).

Again in terms of global and cultural issues, stating norms upfront during the kick-off meeting can be very beneficial:

> So, it, you know, I think it is laid out well during kick-off time and we very seldom have to go back and review it, but if you did have the need it is there for us. So, I think it does help. And especially during, you know, the forming/storming/norming phases because at first they are . . . that is where I have noticed that there is some virtual differences or some cultural differences in the virtual world and you really have to emphasis that you want to hear from everybody in the group, for example. And so, at least stating that upfront so people understand that does help (A01002).

One area in particular where norms can be helpful in a globally distributed environment is communication. It is crucial that people understand what the team needs to working on and what the issues are:

> So in that regard the standard is for everyone, so that everyone understands what you are working on and issues that you have. I think that that no matter what kind of project you work I think it is important to understand you know the communication skills you need and since we are all, almost everybody is located at a different spot. So we are all on the telephone all the time, there is very, very

> little face to face, very little. So it is real important that everybody communicate verbally really well and that we also have written communication (A01001).

In terms of the source of core norms the general feeling of the team was that they came primarily from existing organizational policy and procedure rather than from the members themselves, “I think it comes from the organization. It used to be that it comes from the teams, but these days it comes from the organization” (A01009). Many of the policies and procedures are related to common issues such as “you know, other training that is required. Like stuff like sexual harassment and etc” (A01016).

Due to the corporate nature of the core norms, much of what is done is kept on the technical and performance level. The team does submit status reports on its progress and that is evaluated by the project manager. If there are work- related issues typically the project manager will speak directly with the individual or individuals involved, “so there’s methods and avenues to communicate and work on issues like that. I think that communications is kept more technical in nature. Progress related” (A01010). Due to the globally distributed nature of the team comments were made in regard to the establishment of core norms in this type of setting:

> I think it comes more from the organization, not from the team. We haven’t . . . and the reason why I say that . . . is because we have not had any issue with policies and procedures in the team and our team is located around the world and we don’t have any issues have far as code of conduct or anything like that, everybody pretty much conducts themselves in a professional manner. So, I think it is organizational (A01013).

Specifically related to the use of agile methods within a globally distributed environment, the team must rely to a large degree on the daily Scrum calls to monitor the work that is being accomplished each day. The use of an ICT such as instant messaging is also a way to communicate and verify that the work is moving in the appropriate direction:

> I think it's more the organizational policies and procedures, there is not a lot of thinking of thinking of norms of behavior like work hours and you really don't have a lot of visibility to people's comings and goings so there is not . . . there is not a lot of regulation there you just kind of have to go with what, what stated policies are, the company, and the . . . I kind of think that is what's driving that . . . because, you know, you don't know what folks are working on throughout the day, you have to rely on the daily Scrum calls and their availability as you need people during the day. Are they available, are they getting things done, it's a lot more we rely heavily on instant messaging so that we can kind of know at what rate things are getting done and are things going in the right direction, is the work getting done (A01007).

It is important for the members to be open to the existing policy and procedures and to implement them, so ". . . that kind of atmosphere is definitely established with the team itself and the people who lead it set the tone for that" (A01015).

Team Composition

Size. There was mixed sentiment among the members in regard to team size. Some members felt that smaller teams contributed more to a successful configuration for globally distributed agile teams, while others did not think size was a determining factor as long as strong leadership and management was in place.

In regard to smaller team size, comments included, “Smaller teams are always preferable to larger teams“ (A01003) and “I would, based on my experience, smaller team works great” (A01012). For those advocating smaller size, the following advantages were identified: ease of management and more effective communication, less conflict, greater sense of responsibility and “teamness”. As one member commented, “I think a smaller team is easier to handle and control and make sure everybody, you know, understands what everyone is doing. Easier to keep the small team moving in the same direction” (A01014). Considering the importance of communication within the team often as a team grows in size more people must be included in the communication loop which can increase the difficulty of managing the team:

> I think as you add more people you have more communication touch points, you know I think, you know the ideal team size is something less than twenty, and umm of course there are larger teams as far as teams that have resources for training, you know, and delivering training, which I look at as being completely separate from the task of actually building the product itself. If you have a very large team trying to build a single piece of software it becomes an enormous challenge to keep everybody on the same page (A01003).

In addition to size, task design includes the sub-dimension of meaningfulness, which was addressed early. The size of the team can impact the sense of “teamness” and shared responsibility on the team. As a team becomes larger, members may sense a loss of “teamness” and react by not being as involved in providing input and feedback, and my fade into the background and only worry about their specific responsibilities rather than the well-being of the team overall. As one member on the team indicated:

> But, my experience of . . . once the team gets real large they don’t . . . I don’t know how to explain . . . if you have like a medium team size, say like five people to fifteen, maybe I think there is a lot of real feeling that there is a team involved and that they are all working together, they know each other, they help each other out, I think if it gets too large there is a tendency for people to sit back and focus on their part of the work and maybe the teamwork to drop down a little bit (A01002).

Additionally, due to the distributed environment, the members on a larger team could “see less of a feeling of responsibility and things would not get done . . . you have diminishing returns from more people as opposed to it improving” (A01007). Size, as an element of building “teamness” may also affect the amount of potential conflict on the team:

> But, the idea . . . what I’m, I guess, I am trying to get at is I have found, you know, smaller teams do work really well and even if there is a big project, I do believe that the big project could be broken down into smaller individual teams and have the interfaces managed more than have humongous team with so many more personalities which come through and lead to conflict (A01012).

Size may also have an impact on the speed at which the team can progress. Depending upon the experience level of the members and how long they have worked together may cause the team to slow down if it becomes too large. In addition, having fewer members on the team may help alleviate cultural differences when new member or trainees are brought on the team from other countries. Having a smaller team helps facilitate the relationship building process and new members do not get lost as easily in the crowd. Thus, teams which grow too large:

> . . . could slow down a bit, depending on their experience, whether or not they have worked together before and as you get into larger teams then, then again . . . . I mean if you have people who have worked together, know each other's culture, they . . . they are seasoned veterans at it, it works well after the experience happens, but the start, if you have people from new countries or trainees along with some senior people, etc., in a larger group is does slow the process down (A01005).

Another issue that emerged was the use of full-time and part-time members on the team and the allocation of dedicated resources as it related to team size. With dedicated members, a smaller team would be possible and potentially more successful. As the following member stated:

> I noticed it in the respect of full-time versus part-time. So half my team is part-time and I think my team would be smaller if I could get more fulltime people and I think it would be more successful if I could count on having dedicated resources. In some areas I do not need a fulltime person I can be able to call on someone more occasional consulting so that would not change. I do think in some aspects a smaller team, dedicated team, would be better (A01001).

There were also members on the team who suggested that team size did not make a significant difference in a successful configuration. Comments included, "I don't think it makes a difference" (A01008) and "I haven't noticed that much of a difference. I think larger it is a little hard, but other than that I have not noticed a lot of difference" (A01016) and:

> I haven't noticed that because usually if it is a large size, you know, not the whole the team . . . it is broken up into smaller teams anyway, it does not matter if the teams are large . . . tasks are broken, and smaller teams are formed (A01009).

The concept of reaching a "critical point" in terms of the effectiveness in terms of size was also identified:

> I think it is about the same, I think there is a point at which, a critical point at which a group that is too big starts becoming ineffective. And that may differ a bit, but generally speaking I think once whether its either a team leader, or program manager, or whatever structure the group has so as long as you start having, you know, 6 to 8 people working with you, under you, then it becomes maybe starts to reach that critical point at which it . . . the success of the project may be affected. But up to that, you know, point whether you have one or two people or five or six probably doesn't matter, I think (A01006).

However, stipulations were placed upon the use of large teams. One of those stipulations was that the team be organized properly, managed appropriate, and contain effective leadership. This was indicated by the statements made by the following members. The first member stated,

> No, I don't think it matters if it is lead correctly. I have seen small teams and large teams, I'm talking 100 people, but now, understand, in that 100 people it is all

> parts of people. 10% here, 20% there, 50% there, 30% there, and it's . . . we operate in a very, very highly matrixed environment, but as far as the physical size of the core team, I don't think it matters. I wouldn't say one project is more successful because it had 6 people than another one that had 20. Not at all. I think it goes to the leadership (A01010).

The second member echoed the sentiment in relation to the organization of the team:

> I have not noticed . . . well a large team obviously requires more structure because you have to set up reporting techniques . . . a large team can get out of hand very quickly, it can be very difficult to determine when you have a problem and define it fast enough to fix it before it sinks you on a big team than on a small team, but if they are organized properly I don't see that there is that big of a difference (A01015).

Another element was the amount of work required of the project and the management of the project within a larger team:

> I have seen both, but then sometimes it just depends on the amount of work, but then I have seen both, I mean, even the larger teams having the success, you know. If it is a smaller team it is a little bit easier to control and, you know, manage than a larger team (A01004).

A couple of final considerations in terms of size and team configuration were also identified. First, the fact that sometimes a team needs more members at certain times and fewer members at others was noted. So, there needs to be some flexibility in terms of the size. As one member stated:

> And if you've ever managed projects, I mean, that is really the trend because, you know, there are times when you need a lot of people and there are times when you don't need a lot of people, so that flexibility is really important to maximize the work that is being produced, to optimize it. That's very typical of projects. (A01013)

Second, as mentioned, often a smaller team is easier to work with, but it is important that the team is adequately staffed with members how have the necessary skills to complete the project successfully. In this case "large" may be defined not by a numeric value, but rather by the number of areas or disciplines included on the team, thus, "a larger team meaning, like more areas, more disciplines, different views to the project is better, not necessarily more people of the same thing, so I do not think we need 20 business analysts to the project" (A01011).

Mix. By in large the team felt that having a good mixture of members on the team contributed to the successful configuration of a globally distributed agile team, ". . . the more diversity the better" (A01011). In regard to its importance, one member stated that you "want to be able to, again, foster that . . . that open and honest, collaborative culture in the project. If you have everybody thinking the same way I think that it would be stifling in an agile project" (A01013). Team mixture also impacts the innovativeness of the team in that if members are, "too much alike you not going to be innovative enough and not going learn, be learning as you go, and if you are too diverse, you're going to have a lot of storming for longing periods of time (A01002). From a cultural standpoint, this team was identified as very diverse, "in fact our team was identified as the most diverse team in this region and we did even receive an award a couple months back at the organizational level (A01008).

Interestingly, the emphasis of the responses by members were not primarily related to cultural, personality, age, or gender issues of diversity, but rather centered on business and technical expertise and experience. This may be true because the primary purpose of the team is the development of software applications. As one member commented, “so that is the diversity I think you need is business expertise and technical expertise and marrying those. As far as what country they are from it really doesn’t matter at all” (A01003). In regard to business and technical diversity another member stated:

> It is very important. Cultural maybe not so much, but having the right the mix of people, you know, very detailed-oriented people, and people that are very technically advanced and then other people who, you know, can think and manage differently than others, it’s always been very . . . I have found it very important that people collaborate well with each other with lots of different kinds of experience levels (A01015).

This perspective was repeated in the following statements, “as far as, well, I would say that a technical similarity might affect, but then otherwise all other aspects like you know geographically or culture I haven’t seen any impact from those” (A01004), “I am thinking of it more of a diversity from a technical point of view as opposed to, you know, do you need people from other countries, that is not what I am talking about” (A01016) and “we look at skill sets. Doesn’t matter. It is skill set based. Does not matter” (A01010) and:

I don't focus on that diversity side, I think is an overused word. You bring people together and everyone has a common goal and common set of tasks, you know, that they want to strive for, you know, everyone does their best job, I don't think we necessarily go out and specifically target certain types of people except maybe in terms of technical knowledge they may have (A01014).

Specifically in terms of business expertise, you need "someone who understands or can tell you what the requirements are" and you need a technical person who can identify potential pitfalls in regard to the "processes that are being used to build the software whether that it is configuring an application server, or whether it is trying to figure out how to build and distribute the software, you know, for deployment purposes" (A01003). Because the team never knows for sure exactly what types of problems it will encounter and how much time will be needed to deal with those problems, "if you don't have good technical skilled resources available, you can really get bogged down" (A01003).

So, at the beginning of the project it is good to have "experience in all the necessary requirements that the project entails . . . this ensures that you "have all your bases covered at the outset of a project" (A01005; A01006). Diversity in work experience both managerial and technically speaking allows the team to better understand the client's expectations, "so the more diverse in every aspect, the better" (A01011). Fortunately, "an advantage of the distributed you are not attempting to find everyone who can work in the same room, you can pick people who are strong technically and it doesn't matter where they are . . ." (A01007). However, a disadvantage of the distributed environment was the loss of the mentoring relationship of a colocated team and “the transfer of knowledge that you would get if everybody were together” (A01007). Additionally, “sometimes the time zone presents a challenge” to the way a team can configure the mix of the team (A01001).

Another factor related to the mixture of the team in regard to experience dealt with creating a “very clean division of labor” (A01012). This means that roles are clearly defined for the lead, the architect, the developers, and the testers, “and everybody brings their expertise and their experience into that area, but coming together” (A01012). This would then be topped off by a competent lead “who will be able to listen to multiple viewpoints, but ultimately come to a conclusion. Which, you know, for the most part might not be the general consensus, but might be in the best interest of the project” (A01012).

Thus selection of members comes down primarily to skill set, technical and/or managerial, availability, and location of that person (A01001; A01014).

> Yeah, I think it is a combination, it all depends on the position that the, you know, leads, it is going to be a combination of their business background as well as technical, and with our offshore resources it tends to be, it is sort of the same, I think the technical gets a little bit higher grading than, business experience they might have (A01016).

Technical members were chosen from a pool of resources based upon the specific skill set that they possessed. Managers, leads, and architect are select in a slightly different way. Their selection was based upon expertise and domain knowledge, for example,

> the leaders of the applications area will come and say you have more expertise in this type of application or this type of environment so we are going to assign

> you to that and then there is a small pool of project managers that have a lot of transportation industry knowledge and so those get pulled into the areas that they are considered most applicable for . . . (A01015).

Task and Knowledge-Related Skills. As expected, the team considered task and knowledge-related crucial to a successful configuration. Comments included the following: “I think it is very important, definitely” (A01004) and “I think those are very critical” (A01016). As this team member stated:

> Well, with the different parts of the project you have different activities, different types of skill sets are required, so, you know, each person has their strengths, and that is where we try to put people where there . . . they can provide the most value based on their current knowledge (A01014).

When it comes to team selection the particular skill set of the member was deemed very important. As a team is formed the management may decide that it needs a particular type of developer (e.g., .NET or Java), “and then you find people that are available and then you inventory their skill sets and they may need some training and then we go fill that training” (A01010). However, management may need team members to work on a volatile and critical project which has a high deadline pressure. In this situation, the members selected must already possess the necessary skill set and training is not feasible. Based upon the task levels, the skill set of the members are very critical in the selection process. So, knowledge upfront about the requirements and business needs of the project can help project management determine the types of members and necessary skill sets that are required. As one member suggested:

> I think it is very important especially at the outset of a project, usually cause that where you have to define the requirements and the business needs and requirements of the project, so I think it is very important to have those things covered at the beginning of a project (A01006).

Thus, from a project management perspective having these requirements covered at the outset of the project, “I think that . . . that will . . . you have more chances of a successful project if that is true” (A01006). Having the skills necessary for the task a member is assigned may also build trust of that person among the other members and create a stronger sense of “teamness”. Although business skills and professional skills are important, “I really think that they really need to have the skills to get the job done” (A01002).

Another aspect related to task and knowledge-related skills mentioned by a member was the team role. For example, one member may act as a subject matter expert while other members work as developers. In this particular situation, developers “don’t necessarily need to know how an [industry] operates because their role is to develop the code”, consequently, “their focus is the expertise on how to develop the code the right way and that don’t need to have the knowledge of how an [industry] runs, so it really depends on the role the person is playing” (A01013). This alludes to a clear division of labor as addressed under team composition as one member highlighted the need to bring in “testing expertise, and development expertise and lead expertise. Somebody who can look at the bigger picture type of role is absolutely needed” (A01012).

There needs to be a balance between the use of existing experienced members and the learning process of less experienced members. So, on any given team it was recognized as a necessity to have the appropriate expertise while allowing a reasonable learning curve for new members:

> You need the expert to know how to get stuff done. You have a couple of people that are new on the team, and you know, you need a learning curve or whatever, that is okay, but you need people that are well-established and know what they are doing (A01011).

In an agile enterprise the learning process continues to become more complex as new components and new people are introduced into the team, thus it “will take some blending in and some adjustments to make sure they can get to up to speed and be able to participate as a developer like everyone else (A01005). In this particular team students with Bachelor’s and Master’s degrees are brought in and trained both for a required skill and domain knowledge for the project. This process may take four to five months, so “a lot of importance to the skill of a person, people before they will get onboard with the . . . on the project” (A01008).

Fortunately, a benefit mentioned of having a distributed environment was the fact that very specialized skill sets can be obtained “when you do not have to worry about geography” (A01007). So having the ability to choose from a global talent pool enables the team to compensate for a lack of experience in one or more locations while new members are being trained. However, in a distributed environment collaboration might be more and complicated “so bringing people up to speed is more difficult in this environment, if you have weak links” (A01007).

In regard to compensating for other members on the team, there was a general acceptance of this practice and an acknowledgement that it occurs in most, if not all, teams to some degree, “happens every day. It is a reality of life (A01005). As one team member commented, “yeah it always happens like . . . it is possible that some members compensate, until they bring the other member on track or something . . . yes” (A01009). However, one member did question about how much it currently occurs (A01010) and one other did not think it should be done.

The need for compensation was related to the skill set that a member brings to a particular project, “I don’t think everybody brings the same set of skills to any particular project, otherwise it would be boring (laughter), so yeah, nobody knows everything” (A01006). For a successful project all of the required areas must be covered and it just not “possible to have everybody with the same exact set of skills. So I think it is only naturally that people will be covering other areas that other people don’t have experience in or little experience in” (A01006). As one member commented, “it is a compliment to have different skill sets. It is definitely not all technical – we’ve got a mixture of technical and business people who are not programmers at all (A01001). Thus, business people can help the more technical members with the business issues, “I wouldn’t be running the project if I did not have them – technical people do not have the background of understanding the business side” (A01001). It is not uncommon for the team to help out members who are having difficulties, “I think we all work together, someone is struggling somewhere then we try and help them out and provide value you know, assistance when needed” (A01014). These are examples of the positive aspect of compensation.

Due to financial constraints compensating for other members, helping them out when they are struggling, and developing mentoring relationships can be beneficial to the team, “I think that there is always going to be a case for mentoring sometime, because we can very seldom afford a full team of total experts because of our cost constraints” (A01002). However, the team must be careful that members do not develop the attitude that others will “pick up the slack when somebody is not putting their best effort through” (A01002). So, an important function of the manager was identified as catching those types of situations and correcting them, however, “if you can’t correct it, the weak members need to go, and unfortunately that is part of it you know” (A01015). Although management cannot always be assured that people hired to fill positions will work out as expected, “I don’t think a team will go far unless everybody on the team is pulling their own weight” (A01015). An expectation was that members will seek the appropriate training for areas where they are not as strong, “so whatever your weakness you could polish up on that” (A01011).

Interpersonal Skills. The ability of team members to communicate effectively, express themselves clearly, and establish good working relationships was considered important to the majority of members in regard to a globally distributed agile team, “the better they are able to communicate and express themselves and be understood, the better or more successful the configuration would be” (A01002). Responses ranged from very important, important (e.g., A01004; A01005), kind of important but not required (e.g., A01009; A01015). As one member commented, “I think that it is very important. I have seen some teams highly disrupted by that” (A01016). Another member stated, “it’s definitely important. In order, and certainly if they don’t get along and they show that during whenever they interact and it causes problems, delays in meetings, or whatever, so, it is definitely important” (A01014). Similarly, another member indicated:

> It helps a lot. It’s, umm, I don’t know . . . well, it makes things easier because then you’re not side-tracked by personal issues and you can just concentrate on working versus whether you have interpersonal difficulties, then you spend a lot of time on them and less amount on actually doing the work you are supposed to be doing. So, I think it is very important (A01011).

Good interpersonal skills may allow for more reliance on “informal communication channels and if you don’t then you really need to do a better job of writing down and getting everybody on the same page so to speak” (A01003). One member suggested that if members are “happy with each other . . . it can positively affect the project success” (A01006). On the flip-side, poor team relationships “can negatively affect the whole project and the success of that project” (A01006). Instilling a common goal within the team may foster a strong sense of “teamness” and contribute to a better working environment and greater probability of project success (A01010; A01012).

The relationship between interpersonal skills and effective communication was mentioned several times. In particular one member reflected on the previous attitude toward communication and personal working relationships, “I don’t think you can, you know, in the old style of development world you could just go sit in your cube and just, crap, you didn’t have to come out for a month. It was awful [laughter]” . . . (A01010). This was considered especially important when “you have these global environments and we are on the phone all the time” (A01001). Due to the absence of face-to-face interaction members cannot rely on body language to help them interpret conversation via ICT. Consequently, when speaking with a naturally quiet member on the telephone who is not saying much, “somebody on the other end of the telephone might say . . . think . . . they might jump to conclusions about what the other person is saying (laughter) or not saying” (A01002).

If interpersonal conflicts exists within a globally distributed environment it may potentially lead to leaving members out of the communication loop altogether, because everybody is out of sight, if there’s someone you don’t like or just have a difficult time working with that person, you just don’t call them, you exclude them from conversations [laughter]” (A01007). With the use of ICT such as telephone, teleconferencing, and instant messaging it is easier to get away with and simply avoid for which a member has an interpersonal conflict and so members may “just bypass that person” (A01007). The result is that input from all the members may not be taken into consideration due to these types of conflicts. Ultimately, the ability of members to work together effectively within a globally distributed, culturally diverse team was deemed an important by-product of good interpersonal communication and relationships:

> I think it is very important, again, I think a successful project allows people to be able to feel comfortable in expressing themselves and there has to be, especially, when you have a diverse team of people you need to be cognizant of their background, their culture, and be able to accept their differences and appreciate the differences that they bring to the table (A01013).

It must be noted, however, that a couple of members felt that work could still be completed in lieu of good interpersonal skills within the team, “so, although I think it is kind of important and beneficial, I would not want to say that it is absolutely required" (A01004; A01005). Table 11 provides a summary of the findings for structure for team A01:

Table 11

A01 Team Structure Summary of Findings

Team Structure	Findings
Task Design	
Meaningfulness	Related to the level of effort put forth by the team
	Visibility or criticality of the project
	Privilege to work with new development and/or new technology
	Overall understanding of the project (big picture)
	Less experienced members inclined to put forth more effort
	Personal interest in the project
	Level of responsibility, project management may be more vested
	Pressure caused by deadline or by others waiting
	Creating a beneficial product to the customer
	Potential to be successful
	Respond professionally regardless of type of project

(table continues)

Table 11 *(continued).*

Team Structure	Findings
Autonomy	Foster collaborative environment
	Flexibility in decision-making process and self-organization
	Establish clear responsibilities and assign capable leadership
	Define plain processes
	Foster collaborative environment
	Flexibility in decision-making process and self-organization
	Establish clear responsibilities and assign capable leadership
	Define plain processes
	Manage and plan individually, not be remote controlled
	Ability to organize own work and not be micromanaged
	Flexibility of leadership and non-controlling project managers
	Foster sense of openness and freedom to ask questions
	More rigidity in decision making process for major decisions
	Experience level of the leadership with project and customer
	Feeling that the leadership has a clear direction for the project
	Some projects lend themselves to higher degrees
	Critical vs. streamlined projects impact level of autonomy
	High to moderate of autonomy

(table continues)

Table 11 *(continued)*.

Team Structure	Findings
	Make decisions in relation to creativity
	Level of control and checks and balances
Feedback	Agile practices help in gathering feedback (e.g., daily Scrum)
	“Be upfront, blunt, and candid” (A01010)
	Feedback from work itself, because it is immediate
	Multiple sources are necessary, feedback from all stakeholders
	Reassuring to hear from project leaders
Core Norms	Not project specific
	Organizational policies for confidentiality and handling data
	Reporting, checking in code, walk-throughs, building of software
	Escalation path
	Do not restrict work of team by enforcing norms too heavily
	Influenced greatly by overall corporate culture
	Less important due to professionalism and experience
	No “Ten Commandments” of behavior necessary
	Must be flexible and left implicit, “we are professional”
	Establish norms at project start-up, may be unwritten/unspoken
	“Be on the same page” is important due to globally distribution

(table continues)

Table 11 *(continued)*.

Team Structure	Findings
	Less evident due to lack of physical presence / fewer conflicts
	Assist in the accommodation of different cultures and languages
	Forming/storming/norming
	Crucial that members understand what is expected of them
	Deal more with technical and performance level
	Rely heavily on daily Scrum to monitor work progress
Composition	
Size	Small teams:
	Ease of management
	More effective communication, less "contact points"
	Decrease the potential for conflict
	Greater sense of shared responsibility and "teamness"
	Easier to keep moving in the same direction
	Help to alleviate cultural differences
	Help to facilitate relationship building among members
	Possible if members are allocated 100% to the project
	Should be adequately staffed with necessary skill sets
	Flexibility to ramp up

(table continues)

Table 11 *(continued)*.

Team Structure	Findings
Large	teams:
	Challenge of keeping everybody on the same page
	Less feedback, tendency to fade into the background
	Teamwork my diminish
	Less of a feeling of responsibility
	Increase the potential for conflicts
	Speed at which team can progress may slow down
	Break down into smaller sub-teams
	Must be organized and managed properly
	Must be assigned strong, effective leadership
	"Critical point" where large team becomes ineffective
	Flexibility to ramp down
	"Large" defined in terms of number of areas covered
Mix	The more diversity on the team the better
	Open, honest, collaborative culture
	Diversity improves innovativeness
	Emphasis on business and technical expertise and experience
	Diversity in managerial and technical experience

(table continues)

Table 11 *(continued)*.

Team Structure	Findings
	Loss of mentoring relationships and transfer of knowledge
	Time zone differences may also play a part in mix
	Clean division of labor
	Competent lead to listen to multiple viewpoints/ make decisions
	Technical members chosen based upon skill set, location, and availability
	Project management members chosen on the basis of expertise and domain knowledge
Knowledge And Task-Related Skills	Different activities require different skill sets
	Assign members to provide the most value based upon skill set
	Dependent upon the nature of the project itself and team role
	Upfront knowledge about requirements and business needs
	Required skills from the outset are beneficial to project success
	Trust and “teamness” increased if members are knowledgeable
	Balance between existing and new members
	Complexity of learning process
	Global distribution provides access to very specialized skill sets
	Global distribution may hinder bringing people up to speed

(table continues)

Table 11 *(continued).*

Team Structure	Findings
	General acceptance and acknowledgement that compensation takes place and the need for helping each other out
	Necessity of business skills
	Difficult to hire experts in all areas due to financial constraints
	Encourage members not to develop a lackadaisical attitude
	PM should ensure that compensation is not a detriment
	Seek appropriate training is week in a specific area
Interpersonal Skills	Communicate effectively and express ideas clearly
	Interpersonal conflict may cause delays in the project
	Getting side-tracked with personal issues decreases focus
	More reliance on informal communication
	Less need for written documentation
	A “common goal” may foster a stronger sense of “teamness”
	Cannot rely on body language
	Global distribution makes it easier to leave members out of the communication loop
	Helping members feel comfortable expressing themselves
	Important and beneficial but not absolutely required

Team Virtualness

A01 Extent (Degree) of Virtualness

For this study team virtualness was defined as the extent of virtualness of the team based upon the characteristics of temporal distribution, boundary spanning, lifecycle, and member roles. The following section provides detailed analysis of the team virtualness dimension and its related sub-dimensions beginning with the extent (degree) of virtualness. Based upon the characteristics of each sub-dimension this team represented a high level of virtualness as summarized in Table 12:

Table 12

A01 Extent (Degree) Team Virtualness

Sub-Dimension	Comment
Temporal Distribution	Distributed across 10 cities and 7 states in the United States, one city in India, one city in Australia, and one city in Mexico
	Multiple time zone difference exist
	Overlap in working hours exist between U.S. cities and city in Mexico, but no overlapping hours exist between U.S. cities and cities in Australia and India
	Multiple technologies for synchronous and asynchronous communication and coordination and collaboration

(table continues)

Table 12 *(continued)*.

Sub-Dimension	Comment
Boundary Spanning	Crossed functional, organizational, and cultural boundaries
Lifecycle	Approximately three years for development of new product and then will be an ongoing production support team providing ongoing enhancements
Member Roles	Not uncommon for members to play multiple roles in the team
	Team consisted of some members who were allocated across other teams
	Team consisted of full-time members and part-time members

An examination of the sub-dimensions of virtualness indicated that A01 was geographically distributed across Australia, India, Mexico, and the United States. Within the United States the team was even further distributed across multiple cities. Consequently, the members on the team crossed multiple time zones of varying degrees. Although overlapping hours are present for the United States members and the Mexico members, finding time for all members to meet at the same time was a significant challenge especially having members in India and Australia which represented rather large time differences. In order for the majority of members to meet the time would have to be after 3:30pm just to try and accommodate as many members as possible. Fortunately the team made use of multiple ICT for increasing its capability to communicate across this diverse range of time zones. In terms of boundary spanning the team crossed functional, organizational and cultural boundaries. With members geographically distributed around the world the team identified various cultural boundaries. In regard to lifecycle the team adhered to a longer lifecycle indicating that the development of the project will take approximately three years and will need ongoing support indefinitely. Finally, in relation to member roles, it was not uncommon for members to play multiple roles on the team. The team did consist of members who were allocated across teams or were part-time members. Based upon the characteristics of each sub-dimension this team represented a high level of virtualness.

Temporal Distribution.

Benefits. This team mentioned several of the benefits typically associated with having members distributed in space and time such as a global pool of personnel resources, lower development cost, and twenty-four hour work day, and coverage of the project, “I think the benefits are probably the fact that an employer can utilize the best resources, at the lowest possible cost. So it is more of funding issue and also access to skills that are not available locally” (A01003). Similarly, “I think the only thing that companies see know is, well, offshoring cost . . . “ (A01004). In regard to finding potential members with the appropriate skill sets and expertise one member stated:

> We have wider access of skill sets and there are people that are experts in XYZ that might be in India and we can go to India and get that expertise, or if that expertise is in Japan or Australia or Mexico, wherever it is, we have a broader spectrum and it’s easier to find an expert (A01002)

Another member commented their belief that “one of the main advantages will be that it is cost effective to work in this model and as you know we are able to find the required skill set of people and ramp up the team quite quickly” (A01008).

In terms of the potential for facilitating a 24x7 shop, a globally distributed team has the advantage of a “24 hour coverage of things and things don’t get done at one time, maybe overnight it gets done because it moves to another person” (A01009). The team is no longer ”restricted to an eight or ten hour period, you can increase it to be a 24 hour period, around the clock, in which you have somebody working on that particular issue or problem and so on” (A01006). Thus, work on the project does not have to wait until the next morning, “somebody could pick it up and work on it and leave it, you know, at a more advanced stage when you come in the next day” (A01005). This concept was referred to as the “follow-the-sun if you will, there is somebody working on the project at all times. So, the delivery is quicker” (A01005). This in turn may lead to “faster resolution to the clients and customers, not only within in the project but even, you know, within the problem log situation, when you are trying to fix a problem or something like that” (A01006). This member commented:

> I think nowadays IT is not really a five day, eight hour situation anyway. I think it is more, I have got to get the job done by a certain time and there is lot of emphasis on the customer and client. So, I think it is overall tremendous benefits of having a distribution of people (A01006).

Another member indicated, “oh yeah, totally, I think at least the concept is supposed to make it quicker to delivery and the work is follow-the-sun if you will, there is somebody working on the project at all times” (A01005).

An indirect benefit may be the innovation that is employed “because you do have some challenges, but you resolve those challenges (laughter), and a lot of times very innovation comes into play there” (A01002). Finally, distributed teams allow more flexibility in work location and provide a way for members to be located in places of their choice while still accomplishing the same task as if they were colocated, “anyway, that, you know, would be beneficial in a lot of ways and I would still be able to do the job that I used to do when I was in an office environment in Los Angeles (A01006)

Challenges. A predominant challenge in globally distributed agile teams identified was communication. This ranges from discussion of complex technical and business issues to keeping members informed of policies and procedures as indicated by the following member:

> Being able to effective, you know, communicate complex, technical issues or even complex business issues. And then, just keeping people informed, dealing with stuff like standards, whether they be coding or design standards, so everyone’s, you know, kind of doing things sort of the same way. And not having ten people on a team, doing it ten different ways. (A01016)

Communication is also affected by the time zone difference between locations. Members often do not have instant access to others because the work day has ended where they are located “you can’t just dial-up somebody, oh, I need help from Billy in California, he is only two hours beyond me, I can get him now, you know, that kind of things, it is a bit of a challenge . . .” (A01010). With such a diverse group of people on the team from multiple countries, language is also cited as a common challenge to communication. Simply, can I understand what that person is saying?” (A01015). In response, this team member stated, “I think it is absolutely crucial that the members of your team are able to communicate verbally and on a written level, otherwise it really, really slows you down” (A01015).

Another common challenge of global distribution related to the issue of time zone differences. Multiple time zones often make it difficult to communicate in a timely fashion and to find a time for meetings. As one member stated, “you know, that we have to have more phone meetings and that you can’t necessarily, you don’t necessarily always have instant access if, you know, you can’t reach them by instant messenger or by phone (A01014). Another member indicated:

> . . . if you want quick answers that happens, you are just waiting till like the next day and the person will get your email or you can’t just call and talk to someone or, you know, or send an IM to them. So that . . . that’s a big challenge. If there is a few hours difference, there should be some overlapping hours between the team members. So that is definitely a challenge as far as time wise (A01004).

Common to most globally distributed teams the lack of face-to-face interaction and the use of ICT such as email, Jabber, or instant messaging represents an ongoing challenge, thus, “everybody has to desire or be very motivated to do it. And that is the only way it works. But, everybody has to be put forth the effort to make it work” (A01009).

Finding a reasonable time for meeting was also identified as a recurring challenge, “yes, there are some challenges simply because of the time difference, and the fact that I sometime have to get up at 4 o’clock in the morning for a call” (A01006).

As one member stated, the difficulty in lies the fact that "because we are so global that finding a time for everybody, that somebody has to kind of sacrifice for the team in order to get everyone on the phone at the same time. It is a challenge" (A01001). Echoing those sentiments, another member commented:

> Yeah, I find it more a challenge, because, it is hard to schedule meetings and also the face-to-face communication you can't do, if you are not in the same town and being across continents it's often hard, even it's a virtual meeting still you can't have people get up at three in the morning to attend the meeting (A01011).

Having a member in Australia just means that most meetings must occur from 3:30 pm onwards, which has not posed a problem so much as it just needs to be considered. In addition having members on both coasts means that catching people in Los Angeles occurs after 10:00 am. Offshore resources are located in Mexico and one member reflected, wondered how the configuration of the team would be different if they were for example located in India instead. In answering that question this member responded, "I think it would be very different . . . I think it has been very helpful to at least have them in our right time zone, but for the fact that we never see them, I don't know that that has been a particular problem" (A01007). So time zone differences can be a major drawback, "the time zone is the big difference, and finding a good time zone where we can get all of the team together has had its challenges so that is why I say it is a challenge" (A01013).

This team also faced various challenges beyond communication and time zone differences at varying levels. First, it can be easy to "lose the focus of the overall project and lose sight of the big picture and get little pockets of things where things when you try to bring together don't fit" (A01002). Thus a function of the project management team should be to keep the team headed in the right direction and keep all the members on the same page. A second challenge, although not as predominant, according to one member was cultural differences and diversity. However, the team must remain cognizant that it contains member from various parts of the world and that having "understanding and what they think of it as being professional, you know. It is a little bit different, each places . . . each place" (A01002). There must be an attitude of collaboration and members need "to get used to . . . you got to accept it and trust that it will happen the right way" (A01005). Third, technology at time posed a challenge to the team

> Oh, even with the best of our toolsets, you know, occasionally we are in meetings where we are all supposed to be looking at the same thing and technology just doesn't agree that day, it's not going to work for us, so that is a challenge. It is very important some time that you can visually see something, everybody looking at the same thing (A01002).

Initially, the team encountered connectivity issues on occasion due to insufficient band-width which has currently been solved with more reliable links and high-speed broadband connections. Finally, in a globally distributed agile team it was difficult to gauge how much multi-tasking was occurring among the members that may have an impact on communication and understanding. It may be difficult to know "how much multi-tasking is going on out there, but whether people are really catching it, you know, most meet sometime to have repeat back to you so that your really sure they all heard the same thing" (A01002).

See Table 13 for summary of temporal distribution benefits and challenges:

Table 13

A01 Benefits and Challenges of Temporal Distribution

Benefits	Challenges
Global pool of resources	Communication in regard to both complex technical and business issues
Lower development cost	Time zone differences limit instant access to members and quick answers
24 hour day and project coverage	Language barriers and verbal and written communication skills
Handing off work or "Follow-the-sun"	Communicating in a timely fashion
Quicker delivery	Scheduling meeting times was difficult due to few overlapping work hours
Innovation	Lack of face-to-face interaction and use of ICT
Flexibility in work location	Geographic distance not a problem, but time zone was a difference story
	Loss of focus and seeing the "big picture"
	PM must keep the team headed in the right direction
	Cultural differences and diversity
	Technology - just doesn't work sometime; initially connectivity issues

Use of Information and Communication Technologies. The team has implemented multiple ICT to assist in communication including telephone, instant messaging, desktop sharing, whiteboards, email, and Jabber. Individual calls and conference calls were cited as a one of the primary means for communication among the team, "a conference call we wouldn't be able to function without conference calls" (A01015). Calls are an every day occurrence among members and "sometimes it drives you crazy [laughter]" (A01015). Because dialing internationally can be an expensive option, the team has "a phone system where people are able to, you know, talk easily to one another without the charges being too high" (A01002; A01008).

Instant messaging was identified as necessity in a globally distributed agile team, ". . . because it is hard to sometime pick up the phone and call someone from another country" (A01004). In addition instant messaging is typically used as a first option during overlapping work hours in an attempt to decrease the expense of phone communication, "definitely, the instant message is what we are using the most often because it is cheaper than telephone and is instant, okay" (A01009). Instant messaging was mentioned as the most commonly used tool for scheduled meetings because it "is very cost effective" (A01008). Interestingly, one member commented on how different personalities respond to instant messaging, but still recognized it's usefulness to the team:

> I really do not like IM, but it is a very effective tool. Yeah, maybe it is just my personality, the way it starts flashing at you down in your screen, it's like hey, hey, look at me, look at me, I want you know, talk to me know, it's like I just think

> its rude, I don't like it. But, it works really, really well, because we people really respond to it, much quicker than email (A01015).

Collaboration and desktop sharing applications are used daily for meetings, presentations, and working together on software development. Microsoft® NetMeeting and Microsoft® Communicator are two of the primary technologies utilized as well as an in-house application for presentations (A01002; A01010):

> We have, what it is called, NetMeeting, and Microsoft Office communicator, so, especially the NetMeeting I find quite helpful because you can teach people or show people, help out people, you can see, you know, it is easier that way just to get together and look at one person's screen. I have found that quite helpful (A01011).

Microsoft® Communicator was chosen for its desktop sharing capability considering that the team using this feature extensively, "it's high importance. I have probably been on six desktop sharing different sessions with various people" (A01010). NetMeeting was also identified as a crucial ICT for team collaboration and communication that is used a lot (A01010; A01015). One member stated:

> We use NetMeeting heavily. So probably a day doesn't go by that we don't use that product. We all want to see what is being tested, we all get on NetMeeting and take control and show different aspects of what we are working on. That is probably a very key. We are constantly on the phone doing NetMeeting together (A01001).

Another member commented, "having the ability to use NetMeeting and do whiteboard discussion via the web is just an incredible technology and it's definitely a must in a globally distributed environment" (A01013). Although, whiteboard has not been used as extensively, and one member stated, "I don't know if it's because it is not dependable, or what [laughter]" (A01002).

This is not to say that the team has not had its share of challenges with the technology itself:

> Of course, there are some challenges when I use the same application sharing with the person in India, well the lines are not as high band-width as it is here and I mean, the desktop, panes very slowly, it kind of also . . . you really need to exercise your patience and slow down the speed at which the network band-width allows, but, overall I still think that's a better option than trying to imagine or visualized what the other person is saying (A01012).

The use of ICT may take some personal adjustments on the part of members new to a globally distributed environment. For the most part people adjust fine but, I think there are some people who cannot handle that, and maybe don't use the tools that are out there to be used" (A01005). Another challenge mentioned was that of situations involving "complex business processes that highly distributed teams probably do not perform well. That may change over time as they get better doing certain things, but that is what it seemed like to me" (A01003).

Boundary Spanning. Overall, the members did not feel that boundary spanning had a great deal of effect on the configuration of the team. In fact, two members responded that they have not noticed a significant impact (A01007; A01006). Not to say that there was no impact at all, but "not a great impact" (A01001). As one member noted, “not so much I don’t think, again it because I think the industry has become over the last 15 to 20 years much more professional, in corners, if you call it that” (A01006). Perhaps boundary spanning is felt more in small companies, but in large organizations:

> I think its, again, I think it has changed a lot over the last 16 years or so, because it has become much more of a global environment. Again, I think, back to . . . I think a lot of people are professional now in that sense and they realize that maybe boundaries, but there . . . but, you know, there’s a lot less obstacles to overcoming those boundaries than there used to be. Yeah (A01006).

Functional. Product delivery does involve different functional teams, this is “just something you have to do regardless of whether something is global” (A01005):

> Functionally or organizationally you do that even when you are local or you need something from someone in a different group or some of your work may overlap with them, so, so it is good to have a set standard for communications or “heads ups” if you will that can keep everybody informed (A01005).

Functionally speaking, working for a very large organization of 130,000 people globally, “you are much more aware that there are boundaries and you have to deal with them . . . “ (A01006). Exposure and working within the context of various functional boundaries brings a measure of familiarity and what was difficult the first time becomes seamless over time (A01005). One member also considered the development of a clear division of labor a functional boundary issue, however not to suggest that crossing-over or overlap was not possible:

> When I say clear division of labor it is more like you have a primary role and a secondary role, your primary role is like a developer or a lead, but your

> secondary role would actually quite honestly kind of overlap with many of the others developers (A01012).

This member made the distinction between "productive" overlap and "unproductive" overlap. An example of "productive" overlap would be conducting some level of unit or system test before actually handing the code over to the systems testing analyst. An example of "unproductive" overlap would be "when two people are doing exactly the same thing because one wants to beat the other in trying to develop a project" (A01012). An important function of the lead and project manager would be to evaluate the composition or configuration of the team to see where "productive" overlap may occur. As this member concluded, "crossing over is good, but when it becomes . . . when that overlap becomes almost through its entirety then you need to start worrying about really is this productive" (A01012). One aspect in particular that was pointed out was the issue of part-time members and its impact on team performance:

> the only way that slows everyone down is the person that who is not full-time on a project, but is trying to split their time over multiple projects, that can be . . . can definitely slow you down, especially when you assign them work and, you know, you have to account that they are not working fulltime (A01014).

Organizational. The team it not identify a great deal or organizational boundaries. Currently different "job families" exist and "so they may have a different reporting structure" (A01002). However, this situation is decreasing. There is also a time-tracking system in place that the global regions are working to adopt so everyone is using the same system, but due to multiple copies there have been “little wrinkles along the way” (A01002).

Cultural. Although cultural boundaries still exist in some measure, one member stated, “I think . . . I think also the world is getting smaller so people are more aware of cultural diversity and such”, (A01006). Another member indicated:

> I don’t think it has an effect. We do work with people in Europe, Australia, America, and of course Asia. Different clients and different people in different organizations. But it doesn’t affect our project, I think ultimately we all work toward a common goal and make it work (A01008).

One member mentioned that currently “there are tools that the company has for calibration to different cultures because different cultures do different things” (A01010). Thus, there an environment of cultural sensitivity has been developed due to the global nature of the organization. For example in scheduling meetings, members are provided an opportunity to give suggestions on the time. Depending upon the location and time zone difference this may mean that members do not have meet so late at night and can meet an hour or two early their time. To sum up cultural boundary spanning, a member commented, “so, I think we are very, very sensitive and reactive to those, to those cultural and time zone, temporally spaced differences” (A01010).

Lifecycle

Overall the members felt that a longer lifecycle lead to a more successful configuration of the team. However, some members indicated that, “I think that if you have good dynamics within the team and good communication I think it shouldn't matter" (A01005; A01010). General comments in support of a longer lifecycle included, "I think teams with a longer life cycle have a higher success rate" (A01008), "definitely the team that is together for a longer period of time" (A01004) and ". . . teams that have stable teams, projects that have stable teams that don't come in and out are much more successful than those that don't, I have seen that over and over again" (A01015). A primary reason given for why teams with a longer lifecycle appear to be more successful was the development of interpersonal relationships which contributed to more effective communication and collaboration:

> It is probably long term, is a . . . I go back to the fact that I think interpersonal relationships are very critical, so I think if you give that time to establish then the project is more successful and so, that kind of points to long term being more success than a short term . . . yeah, I am leaning a little bit toward the long term, simply because usually on projects you do work with at least one person you don't know and again I strongly think the interpersonal relationships are very important toward the success of a project. (A01006)

Another member agreed that the longer members work together the "more successfully they become, because they get to know each other, they get to know each other and they can compensate for things the other person does not make or may not be able to apply and all that" (A01009). This development of good interpersonal relationships and better understanding of the people on the team may also lead to more effective communication and identifying strengths, "I'd say number one is you can understand the way they communicate, you could understand their work habits and you

can embellish and key on their greatest assets and make them more productive in the process" (A01013). Following this same line of thought, this team member stated, "having familiarity with the team members, again, enables you to know what their strengths are and be able to use those strengths in development, using those resources optimally" (A01013), thus when you have members moving on and off the team on a regular basis "and working on other projects are allocated in fifty different ways in a month, different projects, then you also do not, you cannot use them optimally because they are thinking of all of their other things they need to do as well" (A01013)..
The following member echoed these sentiments:

> Yeah, definitely, you start to understand each and their strengths, their weaknesses. And more comfortable asking them questions, so you . . . as you build your relationship with people it is easier to work and to . . . there's the storming that typically appears at the start of the project and you don't have to continually go through that, there's comfortable . . . you're comfortable working with that person, it makes it easier. (A01014).

Longer lifecycles may also contributed to greater buy-in to the project and more committed members, "so, you want someone who is committed, who feels that their, you know, have an important part on the team and that would typically be the person who would be on there longer term" (A01014).

Member Roles

Multiple Roles Within One Team. The general feeling of the team was that members playing multiple roles within the same team did not negatively impact on the successful configuration of the team. Overall comments included, “I would say, no affect, I haven’t seen any, you know, major factors on that” (A01004), “I haven’t seen any affect. People play different roles based on the needs of the project. No, I don’t see any negative affects surely (A01014) and “No, no, we do it all the time in other projects that we are involved, but we have the same group of people on working on different projects and we change roles and it does not affect anything” (A01009). Another member stated:

> I think most of us play multiple roles on the project and in the . . . our organization because probably as familiar as anybody with needing to do more and more for the money you make. So, we all do multiple . . . take on multiple roles for the project and it seems to work. It depends on each individual I suppose, but . . . if you have a weak link as part of the chain it kind of weakens the project regardless of whether they have one role or multiples, so I just do not see it as an issue (A01005).

As stated above it was not uncommon for members to play multiple roles within the same team and in some ways this contributed to growth and career development of the members:

> It contributes to the individual career growth and whatever multiple roles are handled it is done in accordance with the career development plan for that particular person and it does help groom the people and set . . . and sort of satisfying the career (A01008).

As long as the roles are balanced, members serving in two roles may actually provide positive feedback to the team, in that they can see that a member can “actually do the development, and also test in its entirety and come up with an entire good product actually motivates teams to who look beyond their primary focus to be able to expand their horizons which is constantly needed" (A01007).

Roles Within Multiple Teams. In regard to members playing single or multiple roles across multiple teams the members believed that this has the potential to have a negative impact on the successful configuration of the team. One positive, however, which was identified which referred to a situation when projects "touch each other" and having members spread across these projects might keep the teams headed in the same direction" (A01011). Other members suggested that the impact may also be influenced by the individuals, the roles they are playing, the number of roles, and the overall size of the team (A01002; A01006). By in large members, however, members considered members playing roles in multiple teams a detriment to the successful configuration of the team as noted by the following comment::

> Yeah, it definitely has an effect because that person cannot be fully committed to one team and they are trying to spread their time over multiple teams and a lot of times they are not available when the . . . one team needs them, because they are spending time on another team. So, you have limited access to them, limited time, and it really affects the project (A01014).

Similarly another member stated, it can be negative if members "are so involved in the other project that they really do not devote a lot of time to the project you are working on . . . " (A01011). In sum, members are being "pulled by difference responsibilities [laughter] and different directions for whatever reasons. That has been a problem, I don’t know if it is related to agile or not, but, I mean, we have had a problem with partial, part-time resources (A01007). As another member recounted:

> I have had lots of experience with that, I have been split amongst three and four and five teams before in the past. It is very frustrating. It’s very difficult on everyone, because you end up, inevitably, you end up getting overloading. It is almost impossible for that not to happen. (A01015)

One member on the team was allocated across three different teams in three entirely different roles in the following allocations: 40%, 50%, and 10%. The member responded to this type of situation by saying, I honestly believe that it is not productive at all” (A01012). Continuing the member identified reasons why this was not a good situation:

> One, people tend not to look at decrease in productivity when you have to keep switching across teams, and objectives, and goals, and what you are doing. It affects you very negatively, because you have now two different PMs looking at their own projects, allocating you in their own projects, want their own projects to go upfront, unless until you have another bigger person out there who is able to control the two PMs or three PMs, then the person who is actually doing multiple roles across different teams get to suffer which brings down, I believe, brings down their morale and in turn brings down, you know, working with other teams, it kind of gets into a negative cycle (A01012)

Based upon this experience the team member suggested that focus “on one team is the best, but two, maybe is okay. As long as it can be defined more by secondary kinds of roles, but going three, or having them all equally important is not a good way” (A01012).

In a globally distributed environment the splitting of team members across teams can also introduce a potential problem. When members who are colocated are allocated across multiple teams, it is much simpler for the project manager to evaluate the time that is being spent on each project. As one member indicated:

> I don't think when they are remote and they're a partial resource, that you can ever get your fair share. Because, because it's remote. And that has been a problem for our project. You know, we really have found that whenever it is possible, you got to get 100% resource because you are out of sight, and therefore out of mind, more often, it is hard to get the commitment, it is hard to get that 30% or 50% commitment, when the person has other demands that maybe coming from across the aisle or the next cube as compared to the demands that are across the phone line, oh yeah, I will be able to makeup their time at a later time (A01007).

Table 14 provides a summary of the findings for virtualness or team A01:

Table 14

A01 Team Virtualness Summary of Findings

Team Virtualness	Findings
Temporal Distribution	
Time Zone	Time zone differences limit instant access to members
	Communicating in a timely fashion
	Scheduling meeting times

(table continues)

Table 14 *(continued).*

Team Virtualness	Findings
	No quick answers
	Should be some overlapping hours
	Time zone differences limit instant access to members
	Communicating in a timely fashion
	Scheduling meeting times
	No quick answers
	Should be some overlapping hours
	Member in Australia most meetings scheduled after 3:30p onwards
	Geographic distance not a major problem, but time zone is a different story, at least we have them in the right time zone
Use of ICT	Telephone, desktop sharing, whiteboard, email, Jabber
	Expense of calling
	IM to decrease phone expense and for scheduling meetings
	Collaboration and desktop sharing applications, e.g., MS NetMeeting and MS Communicator

(table continues)

Table 14 *(continued)*.

Team Virtualness	Findings
	Difficulty with band-width issues across distributed sites sometimes make the use of ICT a little difficult
	Calls for personal adjustments, just some people who cannot adjust to the distributed environment and use of ICT
Boundary Spanning	
Functional	Product delivery involves different functional teams
	More noticeable in large companies
	Clear division of labor
	“Productive” versus “Unproductive” overlap
Organizational	“Job families” may have different reporting structures
	Time-tracking system used organizational wide is a bit inefficient
Cultural	Much less of a boundary since the “world is getting smaller”
	Tools for calibrating to different cultures
	Involving offshore member when scheduling meetings

(table continues)

Table 14 *(continued)*.

Team Virtualness	Findings
Lifecycle	
Short	Some members felt that with good dynamics and good communication it should not matter
Long	Overall team felt that longer lifecycle was more beneficial
	More stable teams are better where members are not constantly coming and going off the team
	Development of personal relationships which contributed to more effective communication and collaboration
	Provides a means for members to get to know each other better and builds personal bonds
	Good interpersonal relationships lead to more effective communication and identifying the strengths of the members
	Familiarity with members enables others to know their strengths and use those strengths optimally on the team
	Contributed to greater buy-in and more committed members

(table continues)

Table 14 *(continued)*.

Team Virtualness	Findings
Member Roles	
Roles Within One Team	The general feeling was that playing multiple roles on the same team did not negatively affect the configuration
	Acknowledgement that it happens often
	Contributes to the growth and career development of the members
	May actually be a positive as members recognize that playing multiple roles enables them to expand their horizons
Roles Within Multiple Teams	Potentially leads to negative impact on the team configuration
	Positive is that projects which “touch each other” and having members spanning across projects the team can better keep headed in the same direction
	Dependent on individuals and specific roles, number of roles, overall size of the team
	Members cannot be fully committed to any one team
	Members pulled in all directions
	Problem with part-time partial resources
	Productivity may suffer

(table continues)

Table 14 *(continued)*.

Team Virtualness	Findings
	May create conflicts between project managers May bring morale down May be okay if second team is defined by secondary kinds of roles Difficult for any of the teams to get their fair share of member's allocation with a remote, partial resource As opposed to colocated members sitting next to each other in cubicles, distributed members can develop an attitude of "I will be able to makeup time at a later time"

Team Agility

Extent (Degree) of Agility

For this study team agility was defined as the extent to which the team aligns with the general values and principles of agile methods as stated in the Agile Manifesto as well as with the practices of a specific method. Overall, members were not familiar with the Agile Manifesto itself, but were acquainted with the general values and principles associated with agile. Both values and principles were adhered to more indirectly and informally. The team had not adopted a formal agile methodology, but rather a modified version of Scrum coupled with additional practices commonly associated with agile.

As indicated by Table 15 there was some question of the extent to which the team adhered to the values of the Agile Manifesto. It was uncertain whether Value 1 and 4 had truly been implemented. It was believed by some that Values 2 and 3 were followed. In terms of the principles of the Agile Manifesto, the team specifically adhered to 6 of the 12 practices. The team was unsure whether the team followed 5 of the principles and one principle was not followed. In terms of Scrum, the team had specifically implemented the daily Scrum (daily stand-up meeting), Sprints (small releases/short iterations, and Sprint Planning (iteration planning). The team also followed additional agile practices which included Test-Driven Development, time-boxing, velocity/estimation, and small releases. Based upon an evaluation of these sub-dimensions of agility this team represented a medium degree of agility.

Table 15 briefly summarizes the team agility dimension:

Table 15

A01 Summary of Team Agility

Sub-Dimension	Comment
Values	For the most part the members agreed that the team strives to adhere to the values of the Agile Manifesto, albeit more indirectly. Overall, the members were not familiar with the Agile Manifesto itself, but appeared to recognize the values as being implemented within the team. Familiarity was found primarily within the project management team.
Principles	The members also indicated that the team strives to adhere to twelve principles of the Agile Manifesto to some degree, again in a more indirect way. Number 6 and 10 were cited as difficult principles to implement within a globally distributed agile team.
Practices	The team had adopted a modified version of Scrum by implementing some, but not all of its specific practices as well as other practices commonly associated with agile methods.

In response to the following question, “were you familiar with the values and principles of the Agile Manifesto before participating in this interview?” member’s comments are summarized in Table 16:

Table 16

A01 Familiarity with the Agile Manifesto

Member	Comment
A01002	Which I am not going to be familiar with.
A01004	Yes, I was just going to say, I am not familiar with that.
A01005	I am not familiar with it. I am a little bit, but not enough to speak to it.
A01006	No, and no to be honest.
A01007	Are they familiar with it, I would say no. I mean a couple of people are familiar with it.
A01009	No.
A01011	Not until you sent them to me.
A01013	No I was not and I really appreciate you sending that.

Confirming the responses to this question one member commented:

> Are they familiar with it, I would say no. I mean a couple of people are familiar with it, but I think we're perhaps it . . . even the project managers are all coming from waterfall, this is their first experience in an agile setup. So, I am not sure they have seen the Agile Manifesto (A01007).

Values

In relation to the values of the Agile Manifesto when asked the question, “does the team strive to adhere to the overall values espoused in the Agile Manifesto either directly or indirectly?” on member responded:

> We are definitely . . . the leads I would say, we have one technical lead who probably has a good five years of experience in agile methods – so we are at the program management learning and then working with our tech support to implement that. So I don’t think some of the programmers as such necessarily see it as directly (A01001).

Comments from individual members are summarized in Table 17:

Table 17

A01 Responses to Adherence to Values of the Agile Manifesto

Member	Comment
A01001	More indirect.
A01003	I am not sure how well we measure up to that.
A01010	Oh I would think so.
A01011	Yes, I do.
A01012	For the most part I would agree with it. Some of them are really subjective.
A01013	Yes, I agree that they do.
A01015	Yes, I do.
A01016	I think most of them, we do. A few of them we probably fall short on.

Table 18 summarizes specific comments in regard to the implementation of specific values of the Agile Manifesto:

Table 18

A01 Extent (Degree) of Agility for Values

Value	Yes	No	Unsure	Extent of Agility
Value 1: Individuals and interactions over processes and tools			X	You know it is the individual and interactions that provide the value not necessarily just using a tool or a process, so I think there is . . . we've figured out to make them all work without going crazy in one or the other. I think with, I am not sure if the manifesto says that, you know, anybody can use any tool, I think that part of . . . that might be part of what this statement means and I do not necessarily agree with that. (A01014)
Value 2: Working software over comprehensive documentation		X		. . . that second one working software over comprehensive documentation that is a nice one, that's a good goal, but I don't think we are even close to that yet. (A01015).

(table continues)

Table 18 *(continued)*.

Value	Yes	No	Unsure	Extent of Agility
Value 3: Customer collaboration over contract negotiation			X	Well, obviously customer collaboration is difficult, since we don't have . . . we really do . . . our client is considered the portfolio group, so I shouldn't say we don't have a client, we don't have a specific client that . . . a paying client that is working with us to develop the product (A01015).
Value 4: Responding to change over following a plan			X	I think the biggest mindset shift is being able to modify the requirements as you go or getting started at the get go and realizing that you really do have enough requirements in your hand to begin. (A01010)

Principles

In response to the following question, “does the team strive to adhere to the twelve guiding principles outlined by the Agile Manifesto either directly or indirectly?” responses are provided in Table 19:

Table 19

A01 Responses to Adherence to Principles of the Agile Manifesto

Member	Comment
A01011	[Number 6] that’s pretty much the only one, because all the others we pretty much follow one way or the other. Yep, so, yeah, we follow almost all of them, yes.
A01012	Absolutely. I can tell you right now number 10, simplicity, the art of maximizing the amount of work not done. And it’s just the fickle, just because of the . . . (inaudible). The other thing, the most efficient and effective method of conveying is face-to-face conversation
A01013	I would say this is the twelve principles of our project

In addition to the general response to the question, several members commented specifically about how the team follows specific principles and/or difficulties with implementing these specific principles. In terms of Principle 1, a difficulty encountered by the team was that the project did not have a “true” customer, therefore the deliver of continuous software was not necessarily required (A01014). However, the team still tried to adhere to this principle as a standard. For Principle 2, the team adhered to this principle within the established change request process that had been implemented (A01014). It was noted that this principle represented a goal to work toward. One member commented:

> . . . but that's a very difficult one to react to because it creates churn and it stops progress if they're are big changes; if they're small changes, yes, not a problem, keep on going. And this team, I think, is real good at absorbing the small changes" (A01015).

Because of the lack of a customer, Principle 3 was also difficult to implement. At the time of the study the team had not delivered any software so to speak, but was working toward having a deliverable of some type developed by the end of each six-week iteration (A01011). As one member stated, "I mean we deliver in terms, you know, at the end of an iteration, we have a new set of software that is functional, but we haven't specifically delivered that into any sort of environment" (A01014). It was suggested that Principle 4 was "easily implemented" (A01015). In this team the business people and developers did work together and participated in the daily scrum together (A01014). It was noted, however, that the distributed nature of the team did pose some communication challenges to the work between developers and business people. Principles 5, 7, 8, and 9 were all identified as difficult to implement (A01015). The team was pretty good in terms of Principle 10 and 11. In terms of Principle 12, one member stated, "it's a nice one to say you are going to do it, but it's hard to make that happen" (A01015).

Although global distribution presented a challenge for the team and appears to be at odds with Principle 6 the team appeared to have adjusted fairly well to the situation. As one member commented, “I don’t think it has been an issue at all. We have an excellent team and good progress is made, communication is good” (A01015). Another member stated:

> Actually I’m not concerned about number six because of all the technology that we have available to us. With the ability to do NetMeeting and the ability to actually conduct meetings globally using whiteboard and things like that I am not sure the actually face to face conversation is as important as many people think. When you have a team of people globally that are focused on a means to an end, it’s a nice everybody in a room in a fact to face conversation, but I do not think it’s essential at that point. Everybody’s dedicated in the project to making it a success. I think you can achieve the same level of participation using technology tools as opposed to a face to face conversation (A01013).

Language barriers and potential misunderstandings were mentioned as challenges the team has faced while communicating via ICT (A01016). In terms of the language issue one member indicated, "I don’t know if it’s been overly hindrance, I thought it might be, but it seems to be working out okay" (A01014). Another benefit was that the majority of the members were located in similar time zones, excluding the member located in Australia. This provided "instant access" to each member during at least a portion of the work day. As another member commented:

> I believe if they were in a time zone that . . . if we were in different time zone, we do have one person in a different time zone in Australia, and that’s probably the most difficult one in that you’re doing everything through email, so you just kind

of have to wait. So that is a definite problem. As long as they are in the same time zone I think you don't necessarily have to have face to face as long as you have instant messaging and instant phone communication (A01014).

Table 20 summarizes the implementation of the principles of the Agile Manifesto:

Table 20

A01 Extent (Degree) of Agility for Principles

Principle	Yes	No	Unsure
Principle 1: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	X		
Principle 2: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage	X		
Principle 3: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale			X
Principle 4: Business people and developers must work together daily throughout the project	X		
Principle 5: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	X		

(table continues)

Table 20 *(continued)*.

Principle	Yes	No	Unsure
Principle 6: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation	X		
Principle 7: Working software is the primary measure of progress			X
Principle 8: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely			X
Principle 9: Continuous attention to technical excellence and good design enhances agility			X
Principle 10: Simplicity - the art of maximizing the amount of work not done -- is essential	X		
Principle 11: The best architectures, requirements, and designs emerge from self-organizing teams	X		
Principle 12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly			X

Table 21 summarizes the implementation of the agile practices within the team including those related to Scrum and others commonly associated with agile methodologies:

Table 21

A01 Extent (Degree) of Agility for Practices

Practice	Description
Sprints (short Iterations)	Six week iterations (A01002, A01003, A01004, A01006, A01007, A01011, A01015, A01016)
Daily Scrum (stand-up meeting)	Fifteen minute meeting each day via teleconference to report on the status of the project and discuss technical issues (A01004, A01005, A01006, A01007, A01010, A01011, A01013, A01015, A01016)
Time-boxing	Used to track where the team is within the project (A01010, A01014)
Test-Driven Development	Writing tests as code is developed (A01014)
Scrum Planning (iteration Planning)	Determining what needs to be completed during each iteration (A01007, A01014)
Velocity / Estimation	Reviewing what the team developed in the last iteration, how much was actually developed based upon estimation and using that information as input into the next planning phase (A01014)
Small Releases	Trying to provide some functionality earlier, maybe not a marketable release but a kind of tangible prototype (A01002, A01006, A01007)

Methodology

Implementation. At the beginning of this project a waterfall approach was being used. After evaluating the project the project management team realized that the team could potentially spend months in analysis without producing any actual software. There was a real need to facilitate and track progress on the project and the "the real, best approach would be to go iterative" (A01012). The project was then broken down into its logical business units and an agile/iterative approach was implemented. In respond to this change one member commented, "so, do the agile principles work well for us with the iterative, I believe so, and I believe that works much better than the waterfall if you would have any such option of this type" (A01012). Another member stated, "have we, yeah, I guess is your question is it better, are we better off that we are agile, are we . . . yeah, I think it has been great" (A01007). One aspect potentially contributing to the perceived positive influence was attributed to the composition of the team itself:

> Yeah, I think it has helped, definitely. Are we there? No, we are not there yet, but . . . the team gets better all the time and keeping the same members of the team has been very, I think, beneficial. We haven't had a lot of turnover, especially amongst the subject matter experts and the architects, they have been on board since the very beginning and if they got pulled out from underneath us and somebody else was inserted, I think that would definitely impact our ability to, you know, to improve (A01015).

In sum, a member indicated, "I think having the team, all of the team, no matter where they are located understand what our values and principles are actually makes the team work better together" (A01002).

One impact of implementing this agile/iterative method identified was the fact that it “it fosters a lot more participation and creativity and as a result I think that everybody has a voice” (A01013). This has created a team that works together closely and “using these principles for an agile project and I honestly think that the iterative approach that we are using is helping to . . . just improve how we deliver a much a better project” (A01013). In addition it has “made the team more communication and teamwork oriented because of there’s more contact than in the typical development world. Like with daily meetings and such” (A01010). In the traditional development process requirements are defined upfront, then comes design, and then it is handed to the developers who work in seclusion for several weeks and then the progress on the project is evaluated. As one member commented, in “agile world just doesn’t work that way. It cannot afford to work that way, we will fail” (A01010).

The practice of short iteration has placed an additional demand on the team and as the following members stated, “you know I think that there’s been an expectation over the years that we can do things in a shorter time frame [laughter] and so we are” (A01003). The member honestly assessed the situation by commenting “but, is it directly associated with the agile approach, I don’t know” (A01003). However, with the focus on producing a deliverable the team is in fact doing things better and faster. The iterative nature of agile may help to take “some of the risks out of the development process by doing a few things early on” (A01003). Thus the team takes take the riskiest first, technically deal with the most, the largest technical risk (A01002). This creates a level of comfort and may contributed to getting the project completed more quickly (A01003). The short iterations also creates “a deliverable that meets the requirements more so than you would in a waterfall, because you are getting feedback and you are more . . . you're able to implement those requirement changes a lot easier than you could otherwise" (A01007). However, another member pointed out the relativity potentially involved with implementing an agile method in regard to short iterations. The team is expected to have something done by the end of the six week iteration. Thus, any requirements not completed must be pushed to the next iteration or taken the out of scope of the project. In regard to this situation the member responded, "I don't know exactly how good that is, because you, you know, you start taking things out of scope just to meet your deadline of the sixth week" (A01011).

> In its time using an agile method, the team has gone through a learning process of what works best for their situation and has had to be flexible in its use of agile but we have offshore or best shore resources doing coding, so we have kind of adapted the process, so that we do a little bit of design ahead of time in the first couple of iteration so that for the next third to fourth iteration we can be one iteration ahead of the programmers, so it is a little bit of a different view of agile methodology and I think as we've been working these past months and seeing kind of what works, and certainly we will have to re-plan some projects – now that we have been doing this for awhile I think it has gotten easier, it is just a different way of looking at it. I think core development and such doesn't change so that is why the programmers do not see it quite as much. But it is how we plan out the project (A01001).

As mentioned in the previous paragraph the team has had to do some re-planning as it has learned how to best utilize agile. This has led the team in some situations to be less “agile” than desired. When iterations have to be extended the team is forced to step back and re-plan subsequent iterations as well as make rearrangements “due to our month to month man-loading and so on. And so there’s a flow-on effect unfortunately which happened, therefore, I think that to me that really tends to put a negative aspect on it” (A01006). In this member’s estimation:

> I think if you have a smaller project, not as big duration, not as big project team, and you have kind of fixed funding up front, and then I can see that there would be some benefit the using of an agile approach as far as the client and customer goes (A01006).

Another member mentioned that having a project plan with too many restrictions may hinder the implementation of agile, “so that does cause some issues in terms of having to track time and assign tasks, I guess, which maybe appears you are not as agile” (A01014).

Benefits. Several specific benefits were identified with the use an agile methodology in a globally distributed environment. The first benefit centered on the ability to “move the risk forward in the project” via iterative phases (A01001). That is achieved by starting deliverables earlier rather than waiting “until your whole project is designed and coded and then have a big surprise at the end” (A01001). As one member commented, “as far as I can see now, in the midway stage it does help to use this method over the traditional method, it does help identify problems much earlier and work quicker on the project” (A01008). By implementing smaller iterations, problems can be discovered much earlier in the development process, “so I am thankful that we are following this process because I wouldn't want to find these surprises at the end so we are definitely seeing the benefits of it" (A01001). Subsequently, "if something is broken you know it earlier than later" (A01004). A second benefit was from the "from the standpoint of getting the status each day" (A01004). Via the daily Scrum members know exactly what others are working on and if they have developed something that can be reused by other members on the team. In addition the daily status reports also lets members know if someone is waiting on them and thus they can prioritize what they are working on. A third benefit has been the emphasis on customer collaboration ensures closer ties to the clients and hopefully will "yield far better products when we get to market" (A01010). A fourth benefit has been "a little bit of improved communication" (A01016). The fifth benefit has been the ability:

> To see and understand what is required in the application center because we first take a broad brush on everything we are going to deliver and then we can do a detailed design on different aspects so that we can quickly understand the architecture of the project (A01014).

Overall, as the following member summarized, "I think the iterative approach is a lot more productive in the long run. It fosters creativity and it allows people kind of think out of the box, maybe do things that wouldn't have done in a waterfall project" (A01013).

Challenges. The members also identified the challenges confronted by the team. Due to the implementation of short iterations the project can sometime move more quickly than in other methodologies and cause a bit of a challenge, "definitely it is a little faster pace" (A01004). This faster pace may also pose some difficulty as members move on and off the team in terms of understanding the scope of the work. As one member commented, “I think it takes a little more time from an iterative approach to grasp what is being done” (A01013). This problem is exacerbated by having members located in multiple locations across the world.

Tied back to the challenge of having members in different time zones, it can sometime be a challenge to schedule the Scrum meetings, “with a globally distributed team it’s hard to find a common time to sort of agree to participate and if it’s a regular meeting it becomes all the more difficult (A01008). It was considered important that members be able to join the daily Scrum since provides regular feedback and project progress tracking. In an effort to help remedy this situation on occasion members would agree to meet at odd hours in order to be a part of the daily Scrum.

Also due to the globally distributed nature of the team the agile method has had to be tailored to meet the challenge of having developers primarily located in distributed locations:

> We can’t follow all the same approach that the agile community ascribes such as designing and coding at the same time, we really have to do it design and then pass that code off, so I guess that is challenge that we have a more detailed design than we probably normally would if we had, if we were following a strict iterative design principles. It would be, you know, design a little, code a little, so that would be a challenge (A01014).

One very specific challenge related to the configuration of the team in terms of lifecycle and agility. This issue had to do with how resources are allocated on a month-to-month basis which involved contractual obligation with the clients and determining how many people would be working for the client in the months to come. Some contracts mandated a minimum number of people to work on a particular area for a particular client, thus creating the necessity to decide upfront who would be assigned. The combination of specific contractual obligations and the implementation of six week iteration periods created the following dilemma:

> . . . everyone of them will crossover at least one physical month and sometime up to three and it is . . . we don't really have any process in place to formally sync that up with our monthly man-loading requirements and it's difficult . . . you don't necessarily know . . . you cannot guarantee that you will have the same people next month that you have this month, you just cannot guarantee that. You can assume that, but you cannot guarantee that. And so, when you have an iteration crossing months and we define all our task and hours and resources by duration, umm, you come to the end of the month . . . toward the end of the month, you find out that you don't have the same person that you have had and you define that person a certain number of hours and tasks within an iteration and it's difficult to reconcile the two (A01006).

One way suggested to alleviate the possibility of not having the same members on the team that were present in the previous iteration was to implement a formula of two iterations equals three calendar months "so that we end iterations, you know, either half-way through a month or at the full calendar end of the month, it just . . . it would be easier at least if we could sync it that way" (A01006). The strategy of month-to-month man-loading and six-week iterations caused a situation where the iteration "could start on the tenth of the month, or the 17th of the month, or the 28th of the month, or whatever" (A01006).

Organizational mindset toward the use of an agile methodology was also identified as a challenge. In regard to this issue one member indicated that at this point and time there has not been a mandated shift from the waterfall development that the organization has traditionally supported to agile. Within the organization there are small pockets of teams implementing agile which are typically lead by an individual or individuals who are familiar with it. Thus, within the organization agile methods are not widely recognized. This member stated, "I think the challenge is getting the organization shifted that way because when you talk about it now there's still a lot of folks that look at you and say 'huh', what are you talking about" (A01010).

A related challenge mentioned was "educating the sponsor in the agile/iterative development approach. Getting them used to the idea that this is how we are going to be developing this project" (A01007). In one situation the team was informed by the sponsor that they were working on the wrong part of the project during the initial iteration, therefore, the team had to go back and begin work on the part that the sponsor was interested in. A consequence of this request was that the team was in jeopardy of losing people resources that it did not want to lose for fear of not getting them back. This was a learning experience for the team which indicated, "maybe this is because we didn't, we didn't, we didn't setup the development cycle in true agile way and it might better if we had" (A01007). So, it was considered very important that the sponsor understood the process of an agile approach and for the team to clearly communicate to the sponsor what it was working on. The upside was that "I suppose it's gone smoother because it's agile, but it has been a painful part of this particular project that we have gone and changed direction like this" (A01007). Plus due to agile the team will have a chance to re-plan in a few weeks and make additional adjustments which would not be possible if they were using a waterfall approach.

Another member on the team identified several additional challenges encountered by the team. First, the planning process in terms of making estimates can be "very difficult to arrive at because of the lack of information that may exist at the beginning of the project" (A01015). The fact that the team often had very few requirements upfront did pose a bit of a struggle (A01010). After the Use Cases are completed it becomes much easier to make estimates in the areas of development testing. Second, testing also posed a challenge. If careful planning is not done the team can . . .

> end up redoing testing throughout the lifecycle of the whole project, whereas in a typical waterfall approach all of the system and integration is done at the end when everything is finished, but because you hope to deliver things then it has to be repeated a lot during the entire length of the project (A01015).

Third, this member reiterated the challenge of communication with the global workplace because "people are scattered everywhere all over the globe and I mean to maintain some kind of impetus and communication, just basic communication is difficult" (A01015). Locations in Brazil, India, and Australia create the challenge of finding suitable times for the daily Scrum and the majority of coding is offshore as well, "so, that presents a challenge everyday" (A01015). Fourth, creating a project schedule was cited. This had to do with the fact the evaluation process for members is based upon earned values and other factors. Because members only get credit for work that has been completed and checked off. Therefore, it is important to build the schedule with short durations. Because of the iterative nature of agile, members may be working on multiple tasks as well as moving from one task to another over multiple days:

> So, nothing really can be marked complete until the end of an iteration, so for a whole six week period you are not marking tasks complete so when we run the earned value tool it doesn't give us credit for any actual hours until the end of an iteration, so our numbers look really crappy (A01015).

Because this information is used by the organization to determine project progress it may appear that there are problems within the project if those numbers look bad. So, a simple thing such as project scheduling becomes much more important because the numbers could be misleading if it is not constructed carefully. Table 22 summarizes the issues surrounding implementation, benefits, and challenges:

Table 22

A01 Agile Methodology Implementation Summary

Issue	Comments
Implementation	Agile fosters more participation and creativity, "every body has a voice" (A01013)
	Made team more communication and teamwork-oriented due to the daily meeting

(table continues)

Table 22 *(continued).*

Issue	Comments
	Expectation that more can be done in a shorter amount of time may increase
	Helps take out some of the risks upfront due to short iterations and regular feedback
	Short iterations create a deliverable better meeting the requirements of the customer
	Decide early how to deal with requirements which are meet within an iteration
	Iteration cycle and month-to-month man-loading should be synchronized
	Organizational overhead should be limited as not to hinder the agility of the team
Benefits	Moving risks forward in the project via short iterations - alleviate surprises at the end
	Getting daily status through the Scrum (daily) meeting
	Emphasis on customer collaboration helps ensure closer ties with clients

(table continues)

Table 22 *(continued)*.

Issue	Comments
	Improvement of communication
	Fosters creativity and out-of-the box thinking
Challenges	Project moves at a quicker pace
	Time zone differences create a challenge for scheduling the daily Scrum
	Need to tailor methodology due to global distribution
	Synchronizing six-week iterations and monthly man-loading
	Organizational mindset, currently no mandate shift to agile from waterfall at this time
	Educating sponsor
	Estimation during planning process due to lack of information upfront
	Testing
	Challenges of communication due to global distribution
	Creating a project schedule which accurately reflects progress to management
	Few requirements defined upfront

Team A02 Description

The following section provides basic information about Team A02 including: size, how members were selected, locations, type of project, how long the team has been in existence, how long the team will be together, familiarity with the Agile Manifesto, which agile methodology and practices have been implemented, what type of agile training the team has received, other projects the team has worked together on, and the success rating of the team. Table 23 summarizes size, selection, locations, and project of team description:

Table 23

A02 Team Size, Selection, Locations, Project, and Duration

Size	Selection	Locations	Project	Duration
9	Need, skill set, availability, merit, experience, knowledge	Three cities in the United States and Two Cities in Brazil	Software development project in communication industry	Short-term team which will disband at completion of project

The team has been in existence approximately one year. From the initial start up of the project the size and membership of the team changed slightly. The team consisted of both part-time and full-time members. By in large the project management portion of the team was part-time, while the development portion of the team was full-time and allocated to the team 100 percent. During initial development the team consisted of 12 members at its peak. Currently there are nine members on the team of which five were from the initial startup team. Three members participated in the study. Table 24 summarizes demographic information about each member who participated in the study while Table 25 summarizes the title, location, time zone, status on the team, and study participation:

Table 24

A02 Demographics for Members Who Participated in the Study

Title	Location	Employment Duration	Present Position Duration	Team Member Duration	Allocation to the team
Project Manager	USCity1	25 yrs or more	1-2 years	1-2 years	Part
Application Architect	USCity2	17-24 years	1-2 years	6 months-1 year	Main
Lead Developer / Technical Lead	BrazilCity1 3-4	years 1-2	years	1-2 years	Main

Table 25

A02 Initial and Current Members and Geographic Locations

Title	Location	Time Zone	Current	Study
Non-dedicated customer	USCity3	GMT -06:00 / CST	X	
Part-time client delivery executive	USCity1	GMT -06:00 / CST	X	
Part-time project manager	USCity1	GMT -06:00 / CST	X	X
Part-time project manager	BrazilCity1	GMT -03:00	X	
Part-time solution architect	USCity1	GMT -06:00 / CST		
Full-time application architect	USCity2	GMT -07:00 / MST	X	X
Full-time application architect	BrazilCity1	GMT -03:00		
Full-time Lead Developer	BrazilCity1	GMT -03:00	X	X
Full-time Lead Developer	USCity1	GMT -06:00 / CST	X	
Full-time Java Developer	BrazilCity1	GMT -03:00	X	
Full-time Java Developer	BrazilCity1	GMT -03:00		
Full-time Java Developer	BrazilCity2	GMT -03:00	X	

Note: GMT = Greenwich Mean Time; CST = Central Standard Time (US); MST = Mountain Standard Time (US); 3-4 hours difference depending on location in US and daylight savings time.

The team has not adopted a specific agile methodology in its entirety. Rather it has begun to implement a modified form of Scrum along with other practices commonly considered to be agile. It is most apparent at the higher levels of the team hierarchy among the project manager and project and technical, i.e., the project management team. As one member stated, “we are agile with a small ‘a’ right now” (A02002).

According to the project manager the team did have a sense that the project was agile. Although the team was fairly familiar with the concepts of agile the members were not specifically familiar with the Agile Manifesto. This team in its current configuration has not worked on any other projects together.

In general the team employed several practices of Scrum including the daily Scrum, Sprints (short iterations of three weeks rather than thirty days), and time-boxing. The team was also using more agile terminology, ”we’re putting our stories in terms of the user perspective and things like that , so, where I guess for viewing our functionality and our requirements more from an agile perspective” (A02001). The team has two formal meetings per week while the other members conduct a daily Scrum via teleconference.

The team as a whole has not had formal training in agile development just an introduction. One member of the team has had formal training and has served as a “coach” for the other members of the team. According to project manager the team has been successful as indicated by the following comment, “oh yeah, I think so. From my perspective, yes, in terms, yeah, of cranking out work and getting things done according to the schedule I think they were successful” (A02001).

A02 Within-Case Analysis

The following section provides a description of the team and analyzes the interview data to explore the response of members in regard to the three dimensions of the theoretical framework: structure, virtualness, and agility. The analysis of each dimension concludes with a summary of the potentially significant findings.

Team Structure

For this study team structure was defined as the overall design of the team based upon the elements of task design, team composition, and core norms. The following section provides detailed analysis of the team structure dimension and its related sub-dimensions.

Task Design

Meaningfulness. In terms of the meaningfulness of the work within the team, one member stated, “I think they will work harder if they find the work to be interesting and they are given challenging goals to achieve” (A02001). Size and visibility were also cited as possible motivators in terms of the work being done, “but the work of each developer was quite recognized and was quite visible, so . . . because it was a small team, people were motivated in putting their best efforts so that we could see their good work” (A02003). Distributed environments may also affect the division of work as indicated by the prior literature and create a lack of “teamness” (Herbsleb & Moitra, 2001). How the work is divide may also affect the team’s ability to communicate and collaborate. In terms of how “work is parceled out”, one member suggested

> I probably would not want two teams that are in two different locations working on the same component or the same feature, anything like that. I think it makes a lot more sense to have single team that’s geographically together having them working on that component, that feature, so that they are having that discussion among themselves in terms of what does this mean, what’s the right way to do

this, how do we test it, that's all pretty much local there instead of having to have those conversation among multiple locations (A02002).

Autonomy. The team considered autonomy an important element of team configuration in that the more autonomy members are given, "the more productive, engaged, whatever you want to call it, they are" (A02001). The affects of allowing members some level of control over work decisions was positive. As one member indicated, "it had quite some positive affects on the team because everyone can give their opinions, suggestions, and it was reviewed and, you know, put into practice" (A02003). Responses to the level of autonomy ranged from high (A02001) to moderate (A02003).

Feedback. Members indicated that feedback is received at all levels including managerial, technical and architectural in terms of performance, technical merits, and solutions (A02003). One member considered it important that members receive "feedback and a lot of interaction from the technical architect or the technical lead on the project rather than the project manager" (A02001). Stories were also cited as a source of primary feedback on project progress (A02001).

Core Norms

The formation of core norms to help the team to deal with communication, cultural differences, and conflicts within the team was considered important. During the initial stages of the project one member stated, "I think that was the main thing, clear communication, language and cultural differences are things that team members could not understand at the first" (A02003). It was suggested that keeping teams together for longer durations of time might be helpful in the formation of these norms (A02003). There were some differences about the primary source of core norms. One source cited was the team itself through the forming/norming/storming process (A02001). Thus, to some degree "it might be good to just let it naturally evolve in terms of how conflict is resolved as long as it evolves to a productive manner, to a productive solution" (A02001). Another source cited was the organization itself via standard policies and procedures (A02003).

Team Composition

Size. In general the team indicated that smaller teams appear to be more effective. Comments included: "my sense is that it works better with a smaller team, as opposed to . . . just adding more bodies to it" (A02001) and "in my experience, small teams are more effective" (A02002).

Mix. Although considered important, in this particular team the fact that a good mixture of members existed was not necessarily due to an intentional effort. Members were selected based upon need and skill set and the team was fortunate that "people brought different abilities to the table and it worked out well for us so we could specialize and give guys different assignments based upon that" (A02001). It was noted that often the mixture or diversity of team was initially based primarily upon need, skill set, and availability. In addition, merit, experience, and knowledge were cited as selection criteria (A02003). Both pros and cons were associated with having too many similar as opposed to having too many dissimilar members. With a diverse team a learning gap may exist and it takes some time to catch everyone up to speed, but at the same this diversity arouses interest within the team. Another member suggested that at times more homogeneous groups form based upon cultural or regional similarities. This member stated, “the diversity factor is important as far as to the independent level” (A02003).

Knowledge and Task Related Skills. In this particular project knowledge and task-related skills were considered important due to the fact that it “had very abstract definition, abstract specification, so the more experienced, senior members had, it helps us to get a better result” (A02003). The project consisted of both junior and senior members so differing levels of experience was brought to the table (A02001).

Interpersonal Skills. Interpersonal skills were considered fairly important in a globally distributed environment. As one member commented:

> You can’t work in a vacuum. You have to be able to talk and generate ideas and work together and, you know, solve problems together, so I think that was pretty important. I would not have wanted to do this with, you know, seven or eight people at seven or eight locations, it was tough enough with three different locations (A02001).

Due to the distributed nature of the team interpersonal communication “was limited to Jabber, or email, or conferences, so the communication and openness, especially

A02002, he had a very positive impact on the team due to his open-mindedness" (A02003). Table 26 provides a summary of the findings for structure for team A02:

Table 26

A02 Team Structure Summary of Findings

Team Structure		Findings
Task Design		
Meaningfulness		Interesting and challenging projects
		Size and visibility of projects
		Two teams in two different locations should not be working on the same component or same feature
Autonomy		Increases productivity
		Increases engagement
		High to moderate level
Feedback		Necessary from all levels
		More important from technical than managerial personnel
		User stories as feedback
Core	Norms	Longer team duration to build interpersonal relationships
		Developed by team via forming/norming/storming
		Natural evolution within the team

(table continues)

Table 26 *(continued)*.

Team Structure	Findings
Composition	
Size	Small, do not add bodies just to add bodies
Mix	Based upon skill set, need, availability, and merit
	Diversity in skill sets, experience, knowledge
	Downside is learning curve
	Upside is arousal of interest and specialization
Knowledge and Task-Related Skills	Differing levels of experience
	Include junior and senior members
Interpersonal Skills	Cannot work in a vacuum
	Must talk and generate ideas
	Work together to solve problems
	Open-mindedness to differences

Team Virtualness

Extent (Degree) of Virtualness

For this study team virtualness was defined as the extent of virtualness of the team based upon the characteristics of temporal distribution, boundary spanning, lifecycle, and member roles. The following section provides detailed analysis of the team virtualness dimension and its related sub-dimensions beginning with the extent (degree) of virtualness. Based upon the characteristics of each sub-dimension this team represented a medium level of virtualness as summarized in Table 27:

Table 27

A02 Extent (Degree) of Team Virtualness

Sub-Dimension	Extent of Virtualness
Temporal Distribution	Distributed across two cities in the United States and two cities in Brazil Time difference was 3 to 4 hours depending on U.S. location and daylight savings times Approximately five to six hours of overlap in work hours Multiple technologies for synchronous and asynchronous communication and coordination and collaboration
Boundary Spanning	Crossed organizational and cultural boundaries

(table continues)

Table 27 *(continued)*.

Sub-Dimension	Extent of Virtualness
Lifecycle	Shorter lifecycle
	Changes in the members of the team, five original members still active
	Will disband when project is completed
Member Roles	Not unusual for members to play multiple roles on the same team based upon who could do the best job, regardless of job title
	Development members were allocated 100% to the team
	Certain members of project management team allocated across multiple teams

An examination of the sub-dimensions of virtualness that although A02 was geographically distributed across two cities in the United States and two cities in Brazil a reasonable amount of overlap in working hours was present for conducting synchronous communication such as teleconferences, instant messaging, and desktop sharing. Also a benefit was the use of many different types of ICT.

In terms of boundary spanning the team crossed organizational and cultural boundaries. Consider its size of approximately 130,000 employees world-wide, it was cited that more organizational boundaries were present than perhaps in smaller organizations. With members geographically distributed in two countries cultural boundaries were identified.

In regard to lifecycle the team adhered to a shorter lifecycle due to the fact that it will disband at the completion of the project and resources will be reallocated. The team had only been in existence approximately one year and there were changes made to the team from initial formation to its current form.

Finally, in relation to member roles, it was not unusual for members to play multiple roles on the team as a more egalitarian approach to management was employed. The development members were full-time, 100% allocated. The project management members were half-time to the team and were allocated across teams.

Temporal Distribution

Benefits. One of the benefits cited was the cost savings afforded by the use of global distribution. As one member commented, “originally, it was going to be done all with programmers in the United States and when they compared the cost models between the two they said okay we need to go with the offshore team” (A02002).

Challenges. The members cited time zone differences, communication, and language barriers as significant challenges. One member in particular commented:

> I would say the biggest one was really just setting up meeting times when you have so many time zones and have Brazil, you know, sometimes it is four hour different, and then sometime three hours, and sometimes two hours and to me it

> was just the time zones of trying to get . . . find a time where everybody was around (A02001).

Helping to alleviate this challenge to some degree with the team in Brazil was that there were some overlapping hours unlike with the team in India where no overlapping time existed. As one member stated, “I think the time zone thing, you know, could be an issue, you definitely have more time to collaborate if . . . even if you are distributed, but are in fairly equal time zones” (A02002). Due to the temporal distribution of this team daylight savings time also played a role in its ability to communicate and collaborate:

> You know, I did feel like once we both came off of the daylight savings time, because their daylight savings time is working in opposite of ours right, so once we came off of that where we did have about five to six hours of time when we were all in the office together I did feel that there was a lot more collaboration going on together than there has been, you know, once we have gone back to four to five hours difference between us (A02002).

It was noted that with the South American team some of the members were not as fluent in English as others thus creating “a big communication gap that people generally have and communicating on the requirements and specifications and such.” (A02003). See Table 28 for summary of temporal distribution benefits and challenges:

Table 28

A02 Benefits and Challenges of Temporal Distribution

Benefits	Challenges
Cost savings	Time zone differences
	Communication
	Language barriers

Use of Information and Communication Technologies. The team implemented several common ICT. Like most globally distributed teams teleconferencing and email is used on a regular basis. The team has also adopted groupware and collaborative software for desktop sharing and instant messaging. Specifically, Microsoft® Communicator was adopted for these purposes (A02001; A02003). The team has also created a shared code repository as well as a shared requirements repository using Borland® CaliberRM. The shared requirements repository has been helpful in the sense of having . . .

> everybody working from the same repositories of here's where the current version of the documents are and all that kind of thing, the version of the requirements, all that, so that each team, you know, even though they are distributed they kind of know where to go for the gold source of here's where the code is, here's where the documents are, here's whatever is (A0202).

Although not necessarily directly related to communication itself, the team did put together a digital photo album of all the team members so that when involved in a teleconference, chat session, or simply emailing each member had a visual image of the person on the other end. As one member commented on the helpfulness, “I know that when I am talking to people on the conference calls I normally go back to those photos, as kind of I am thinking in my mind as I am listening them talk” (A02002). At the time of this study the team had not experimented with videoconferencing due to lack of necessary technologies or facilities.

Boundary Spanning

Functional. No specific functional boundaries were identified.

Organizational. Members are employed by the same organization in all distributed sites. This has helped to alleviate any major organizational boundaries due to the fact that everyone has “the same guidelines, same working patterns and same best practice” (A02003). As one member commented:

> Yes. I think it definitely helps in that we are all speaking the same [organizational] language. We all know what the current corporate initiatives are, what the recommended tools are, all that kind of stuff. I think it would be a lot harder if we were in different organizations and you know they had a different set of tools versus what we wanted to use in the States. I can imagine a lot of turf war kind of things happening (A02002).

However, due to the hierarchical structure within the U.S. team and the Brazilian team some issues related to responsibilities has been encountered. For example, although members are all in the same organization and may work in the same technical area, Brazilian members report via the Brazilian hierarchy and U.S. members report via the U.S. hierarchy.

Cultural. One cultural boundary identified was in reference to vacations and holidays in Brazil, where team was “a bit more relaxed as far as vacations go and holidays go . . . we have lots of holidays, so that does put a certain barrier on the rhythm or tempo that we have on the project” (A02003). The language barrier was also cited as a cultural boundary, but one that the team was able to work around.

Lifecycle

There was a mixed response to the issue of team lifecycle. One member advocated a shorter lifecycle, “I think shorter . . . the end is near. They, there is more visibility to seeing their work implemented, so I think is shorter is better”. (A02001). One member suggested in the middle, “I would say a medium, not a very long or a very short, because in the short we are still learning about each member I believe initially” (A02003). This allows enough time for the member to get to know one another and to build up steam and increase productivity. Another suggestion was to begin with a small number of members and building up the core group until they are working well together and “going through the practices, then bringing on individuals where you have already kind of established that environment and that culture and that set of techniques that seems to work better than trying to bring everybody in all at once" (A02002). Thus, a more progressive lifecycle is put into place. Turnover then becomes more of "staffing and what people needed when" (A02002).

Member Roles

Multiple Roles in One Team. As far as members playing multiple roles in the team, one member suggested that it might be related to the individual person as to whether it was an issue or not:

> I think sometimes we can ask people to wear too many hats and some people do not work well in that environment and some people can wear a couple of hats, but they have a limit to what they can do, so I think that is a consideration, that is something that needs to be considered, especially where, you know, small teams where we need people to do a lot of different teams, that needs consideration. (A02001)

It might be possible to have a member allocated 50% as a tester and 50% as a developer. Another consideration noted was for those members who role might be more involved toward the latter part of the iteration as is the case for testers. The importance of managing these types of roles was to emphasize to the members that they are a "fully functional, participating member of the team just like anybody else" (A02002). A suggested way of alleviating any problems with these types of roles was to make sure and involve them from the beginning of the project perhaps in the area of requirements gathering. It is important to think about how roles impact the team. As one member recalled:

> Some of that is just based on my past experience where I have seen that being very strict about roles I found to be counter-productive on an agile project. You know, I have found it to work better when you try to create kind of egalitarian type of culture, where everybody's pulling the cart together, we need everybody to, you know, keep us going in the right direction (A02002).

By and large the attitude of this team was "who's got the expertise and do we as a team agree on this is the right way to" (A02002). Thus, the approach taken was more egalitarian rather than based upon the specific title of individual members.
So, it was not necessarily unusual for members to cross roles based upon who could do the best job, regardless of job title.

Roles Within Multiple Teams. In regard to playing roles within multiple teams it was suggested that this was somewhat dependent on what the additional assignments will entail. Do the roles involve high-level or low-level tasks? Ultimately, "I think probably it depends on what the additional roles are" (A02001). Another consideration was related to the individual member. It "depends on the capabilities of the individuals on whether they can handle that or not" (A02002). In this particular team it was difficult to evaluate this issue in depth considering that all members, except project management, were allocated 100% to the project. Table 29 provides a summary of the findings for virtualness of team A02:

Table 29

A02 Team Virtualness Summary of Findings

Team Virtualness	Findings
Temporal Distribution	
Time Zone	Difficulty in scheduling times for meetings
	Need for overlapping work hours
	Daylight savings time should be taken into consideration
Use of ICT	Teleconferencing and instant messaging for synchronous communication
	Groupware and collaborative software for desktop sharing
	Shared code and requirements repository; Digital photo album
Boundary Spanning	
Functional	None identified
Organizational	Using same guidelines, working patterns, best practices
	U.S. team members and Brazilian team members report via separate hierarchies
Cultural	Differences in holidays and vacations
	Language barrier

(table continues)

Table 29 *(continued)*.

Team Virtualness	Findings
Lifecycle	
Short	Opportunity to see work implemented
	Progressive lifecycle, bring members on over time
Long	Allows time for members to get to know each other better and develop personal relationships which may contribute to better communication
Member Roles	
Roles Within One Team	Dependent upon individual
	Involve all members in some way throughout the project
	Strict adherence to roles can be counter-productive
Roles Within Multiple Teams	Dependent upon roles played and individual

Team Agility

Extent (Degree) of Agility

For this study team agility was defined as the extent to which the team aligns with the general values and principles of agile methods as stated in the Agile Manifesto as well as with the practices of a specific method. Overall, members were not familiar the Agile Manifesto itself, but were familiar with the general values and principles associated with agile. Both values and principles were adhered to more indirectly and informally. The team had not adopted a formal agile methodology but had implemented a modified version of Scrum as well as several practices commonly associated with agile.

In terms of values the team was somewhat limited by the organizational culture which emphasized “processes and tools” and “more documentation”. Due to pragmatic reasons, however, the team was able to implement this particular value. Also limiting the team from being more agile was the fact that the customer was not located on-site and thus a customer proxy was utilized. The team did agree on “responding to change over following a plan” due to the frequent changes requested by the customer. The team attempted to adhere to the majority of the principles of the Agile Manifesto; however, two specific principles were identified as difficult to implement: 4 and 10. Due to the distributed nature of the team principle 6 was not possible. The team did implement several practices from Scrum (e.g., daily Scrum meeting and Scrum planning) as well as other practices commonly associated with agile methods. Based upon an evaluation of these sub-dimensions of agility this team represented a medium degree of agility. Table 30 briefly summarizes the team agility dimension:

Table 30

A02 Summary of Team Agility

Sub-Dimension	Comment
Values	Although the team was not explicit about the values for the most part the team strove to adhere to the values indirectly
Principles	The team adhered to the principles indirectly, with Principle 4, 6, and 10 cited as difficult to implement within the team, otherwise, to some degree the other principles are followed
Practices	Modified version of Scrum combined with other commonly recognized agile practices: short iterations (Sprints), estimation, test-driven development, user stores, continuous integration, automated testing, pair programming, daily stand-up meetings (daily Scrum), iteration retrospective, iteration planning (Scrum Planning)

Values

In response to the question, “does the team strive to adhere to the overall values espoused in the Agile Manifesto either directly or indirectly?” one member responded, “definitely we are not explicit about the values” (A02002). Table 31 summarizes specific comments in regard to the implementation of specific values of the Agile Manifesto:

Table 31

A02 Extent (Degree) of Agility for Values

Value	Yes	No	Unsure	Extent of Agility
Value 1: Individuals and interactions over processes and tools	X			Corporate culture favored processes and tools while team emphasized individuals and interactions (A02002)
Value 2: Working software over comprehensive documentation	X			Project management would like to see more documentation; less documentation more of a pragmatic necessity because of time limitation (A02002)
Value 3: Customer collaboration over contract negotiation	X			Difficult to follow since customer not on-site at either the project management location or the development team location; a proxy customer has been implemented (A02002)
Value 4: Responding to change over following a plan	X			Agreed upon by the project management and development teams; also followed on the basis of pragmatic terms because customer made frequent changes (A02002)

Principles

In terms of the principles of the Agile Manifesto the team has took a more informal approach as indicated by the following member:

> . . . and again the principles would be just like the values, I have never put these forward to the team and . . . although I think there would be more general buy-in from the whole teams on the principles more so than the values, which is kind of interesting (A02002).

This member questioned whether there would be much difference to enacting the principles within a globally distributed environment as opposed to a colocated environment, except for the specific principle of having an on-site customer (A02002). In general the team tried to adhere to the principles of the Agile Manifesto so rather than commenting on every principle, the members being interviewed directly for this information chose to respond only to those principles that are difficult to follow in a globally distributed environment.

It was cited that Principle 4 was very difficult to follow since the customer was located off-site and the business function was implemented through a customer proxy (A02002). Specifically in relation to Principle 6 one member agreed that general concept of face-to-face communication was most likely more efficient and effective, but since it was not an option for this team, the member indicated that the approach taken by the team to enact this principle of communication was to limit the amount of paper “shuffling” between the members. Because of the scheduling constraints the team just did not have time to pass design documents back and forth making comments at a distance. Rather than following an “over-the-wall” approach collaboration should be done in real-time utilizing conference calls, instant messaging sessions, etc, “so, I think that is probably the more important principle that I have tried to put in place when we are more talking about globally distributed teams” (A02002). There was uncertainty about the adherence to Principle 10 because so much of it is based upon the buy-in of the individual (A02002). Table 32 summarizes the implementation of the principles of the Agile Manifesto:

Table 32

A02 Extent (Degree) of Agility for Principles

Principle	Yes	No	Unsure
Principle 1: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	X		
Principle 2: Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage	X		
Principle 3: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale	X		
Principle 4: Business people and developers must work together daily throughout the project		X	

(table continues)

Table 32 *(continued)*.

Principle	Yes	No	Unsure
Principle 5: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	X		
Principle 6: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation	X		
Principle 7: Working software is the primary measure of progress	X		
Principle 8: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely	X		
Principle 9: Continuous attention to technical excellence and good design enhances agility	X		
Principle 10: Simplicity - the art of maximizing the amount of work not done -- is essential			X
Principle 11: The best architectures, requirements, and designs emerge from self-organizing teams	X		

(table continues)

Table 32 *(continued)*

Principle	Yes	No	Unsure
Principle 12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior	X		

Practices

In regard to agile practices the team has moved into a modified version of Scrum implementing some but not all of its specific practices. The approach has been to “avoid using too much of the terminology on people” and to “suggest practices to the team, but not require them for the most part” (A02002). These practices have been implemented via white boards, desktop sharing, and remote desktop. Table 33 summarizes the implementation of the agile practices within the team including those related to Scrum and others commonly associated with agile methodologies:

Table 33

A02 Extent (Degree) of Agility for Practices

Sub-Dimension	Description
Short iterations	Three week iterations (A02002, A02003)

(table continues)

Table 33 *(continued)*.

Sub-Dimension	Description
User stories	Story-based planning (A02002)
Estimation	Lighter estimation approach based upon stories combined with corporate templates and methods for compliance purposes (A02002)
Test-Driven Development (TDD)	Implemented from start of project, but optional (A02002)
Continuous integration	Every day code is integrated back into the code base (A02002)
Automated testing	Applied to the pieces done using TDD (A02002)
Daily Scrum (daily stand-up meeting)	Conducted informally from the beginning, often conducted among the members in Brazil (A02002)
Pair Programming	Utilization varied (A02002, A02003)
Iteration retrospective	Done a little bit, perhaps half of the iterations had a formal retrospective (A02002, A02003)
Iteration planning	Activities were planned for each iteration and then activities for the next iteration were adapted based upon the feedback from the previous iteration (A02003)

Methodology

Implementation. According to one member of the team this project was the first time that either he or the project manager had implemented agile within a globally distributed team. Thus, they were not exactly sure what to expect or how things were normally supposed to work. However, this member commented:

> I would say that they overall at least for a project like this one, where there is a lot of changing of direction by the client that agile has been a net benefit over the way we normally we would do projects which is more of a waterfall approach because actually even iterative is still getting a foothold in [Organization A], we don't have many project managers that feel comfortable with iterative. So, you know, agile is kind of really pushing the envelope in some ways (A02002).

As indicated by one member, the level of agile incorporated into each project is driven by what the project manager decides to do. Thus in an overall effort to show managers and leaders the benefits of agile, ". . . we are incorporating agile practices more and more . . . " (A02003). In terms of whether the development portion of the tem recognizes that they are using an agile method rather than simply completing an assigned task this member responded:

> Yes . . . so that way the team works quite independent in the sense not task oriented but they are participating in the decision-making and the tasks and defining the definitions. So they have been seeing the benefits of agile practices first hand (A02003).

Benefits. In reference to the benefits of agile in a globally distributed environment one member commented:

> I would say we have seen quite some benefits coming out of agile practices. Especially better communication and more trust within the team members. Each one got to know the limitations and the best practices of the other developers, so that was very helpful in bringing the team together. (A02003)

Another member identified a “main benefit of agile in that sense I would say is that the development team seems to have more of a sense of ownership of the program . . .” (A02002). Additionally, “in this particular project, agile has been very important because the client is constantly changing their direction based on marketing concerns” (A02002). Flexibility has been another benefit of agile in regard to specifications, “so it was very agile to develop a particular component and get the user feedback and modify it further or go to the next component, so it was quite flexible on that part (A02003).

Challenges. As one member stated, “definitely, there have been lots of challenge in trying to do it [agile] with the distributed team. For many of the U.S. members they have never met the development team [Brazil] in person. Due to some language barriers the daily Scrums initially posed some challenge. However, there are members in Brazil whose English and communications are really good, so often the daily Scrum was facilitated by the development team in Brazil themselves:

> So they can speak in Portuguese and, you know, in a language and in an environment they are more comfortable with. I feel like the meetings are more productive if they don’t feel like they have to conduct those in English (A02002).

In addition to the daily Scrums, two standing meetings were scheduled during the week and conducted in English. In order to alleviate the language barrier those on the development team fluent in English primarily participate and then “I believe then they have side conversations afterwards if there is any confusion or questions” (A02002). In terms of challenges that arise the team tries to take a pragmatic perspective:

> We are more looking at what’s working and what’s not working and we’re trying to look at that question before we start looking at corporate standards or agile playbook or that kind of thing. We’re just really looking more at it in terms of what works and what doesn’t work (A02002).

Finally, there are some organizational constraints “that are put on the team as to how they can go with being kind of autonomous about how they do their work” (A02002). See Table 34 for summary of issues related to implementation of agile methodology:

Table 34

A02 Agile Methodology Implementation Summary

Issue	Comments
Implementation	Net benefit
	Pushing the envelope
Benefits	Agile has improved communication and trust
	Agile has increased a sense of ownership of the program
	Agile has allowed for quick changes to customer requirements
Challenges	Language causes some difficulties in communication during the daily Scrums

(table continues)

Table 34 *(continued)*.

Issue	Comments
	Allow distributed members to speak their own language during daily Scrums to communicate with those who are not as literate in English
	Allow for a brief meeting after the daily Scrum for the team members to communicate and clear up any questions or confusion

Organization B Cases

Organization B Description

Organization B is a United States based company that employs approximately 40,000 employees and outsources the majority of its internal customer application development to a contracting company while keeping project management tasks onshore. In the past its primary focus has been on North America, but with the growing interest in the global economy it is beginning to take advantage of opportunities abroad as well. Organization B has also traditionally held to a structured, waterfall approach to its development. However, recently several informational sessions discussing the values, principles, and practices of the agile methodology have been offered to its employees and interest in their use has grown.

Team B01 Description

The following section provides basic information about Team B01 including: size, how members were selected, locations, type of project, how long the team has been in existence, how long the team will be together, familiarity with the Agile Manifesto, which agile methodology and practices have been implemented, what type of agile training the team has received, other projects the team has worked together on, and the success rating of the team. Table 35 summarizes size, selection, locations, and project of team description:

Table 35

B01 Team Size, Selection, Locations, Project, and Duration

Size	Selection	Locations	Project	Duration
9	Need, skill set	One city in the United States and One city in India	Transportation-related application	Long-term

This is an ongoing project. The application was used on a daily basis and will always be in need of support. Recently a new initiative has been proposed to add some new features to the application. So the team will continue for the foreseeable future. The application itself has been in existence for approximately ten plus years. Two of the organizational employees had been on the team for approximately five years. The contractors rotate on and off the team depending upon budgeting and need. All three of the organizational (onshore) members participated in the study. Members from the contracting company were not accessible for the study due to difficulty of obtaining non-disclosure agreements. Table 36 summarizes demographic information about each member who participated in the study while Table 37 summarizes the title, location, time zone, status on the team, and study participation.

Table 36

B01 Demographics for Members Who Participated in the Study

Title	Location	Employment Duration	Present Position Duration	Team Member Duration	Allocation to the team
Manager	USCity1	9-16 years	1-2 years	1-2 years	Main
Senior Software Developer II	USCity1	5-8 years	5-8 years	3 years or more	Main
Technical Lead	USCity1	5-8 years	5-8 years	3 years or more	Main

Table 37

B01 Current Members and Geographic Locations

Title	Location	Time Zone	Current	Study
Manager	USCity1	GMT -06:00 / CST	X	X
Senior Software Developer II	USCity1	GMT -6:00 / CST	X	X
Technical Lead	USCity1	GMT -6:00 / CST	X	X
Contractor(s) IndiaCity1		GMT +05:30		

Note: GMT = Greenwich Mean Time; CST = Central Standard Time (US); 11.5 hour difference.

Although the teams participating agreed that they feel they are following the overall values and principles of the Agile Manifesto in a general sense, they have not adopted and implemented any type of specific agile methodology, but have begun emphasizing the use of short iterations, iteration planning, on-site customer, small releases, and continuous integration. As one team member put it when shown the Agile Manifesto, “oh, that’s what you mean by agile, yeah, we been following these ideas for some time now” (B01003).

The team has not implemented a specific agile methodology, but follows the agile values, principles, and practices in a more general, high-level approach. The team conducted regular meetings with the customer to discuss requirements and their needs and built demos for showing the functionality of a piece of the application. Customers then suggested changes and the team implemented those changes in the next iteration. Updates are made on a regular basis and one of the primary reasons for adopting this more agile approach was to facilitate customer ownership of the changes and enhancements to the system.

Considering the fact that the team had not implemented a specific agile methodology, the terminology commonly associated with the use of agile did not come out in the interviews. However, in speaking with the members, several concepts associated with agile practices which could be interpreted as short iterations, iteration planning, on-site customer, and small releases were mentioned.

Approximately a year ago a presentation was made on agile development and was discussed within the department and the team was like “oh, that’s what we do and that is kind of how it came about that we were doing agile development” (B01001).

Overall the members were not familiar with the Agile Manifesto, so their adherence to the values and principles was informal and indirect. However, after seeing the Agile Manifesto the members acknowledged that they were familiar with the concepts themselves to some degree.

The team has not had formal training in agile. It was more of an introduction to the concepts. In terms of this specific team they are assigned to one specific project. However within this project there may be application related projects which may include interaction with another team. So, all work was related to this one project. As far as the manager knew the team had not worked on any other agile development projects.

According to one member the team “has been very successful” in terms of interaction with the customer on a regular basis and motivating them to take ownership of what the team was delivering. In sum this member indicated:

> So, I think it has been very successful. It works well in our team, but I know in other teams I have been on or that I manage as well, it is not always effective, because of the big changing requirement and big changing in direction and trying to keep it to a scope, a reasonable scope (B01001).

B01 Within-Case Analysis

The following section provides a description of the team and analyzes the interview data to explore the response of members in regard to the three dimensions of the theoretical framework: structure, virtualness, and agility. The analysis of each sub-dimension concludes with a summary of the potentially significant findings.

Team Structure

Task Design

Meaningfulness. In this team the idea of meaningfulness was not limited to the significance of the work to the team itself, but also to the importance that others in the organization attribute to the project. So, it is not simply whether the work is meaningful to the team but also if the work is significant to the stakeholder as well, as reflected in the following comment:

> I don’t know if having, being part of a larger project would have much to do with it. I would think if you know in your heart that this is not . . . if this is something that is not going to be used versus something that is going to benefit someone I think that would make a difference. If I am about to deploy something that I know no one is going to use it and no one cares about it and the guy who requested it

> retires and then the new guy says, I really don't care about that, that is a de-motivator (B01003).

Autonomy. Having a sense of autonomy or freedom to be creative appears to be important in this case. As one member commented, it has to do with the "well being of the team" (B01003). This member continued:

> Because I think when it comes down to it, you, I don't know, the . . . my experience is, if you give me a team that lets you be creativity and have input and you feel that things get part of the team versus one that's dictated to, I think the prior one is a better makeup (B01003).

The level of autonomy was somewhat predicated on the experience level of the team members (B01002). Those members who have been on the team longer typically would have a greater degree of autonomy than newer members of the team. In reference to the offshore members of the team the following comment was made:

> We truly try to celebrate them offshore when they come up with creative ideas or ways of handling something that maybe we have not thought of. So we want to encourage them to always bring those ideas and things up and yet in the same sense again, you know, depending on their experience whether they have the autonomy to implement those changes or it needs to be reviewed first (B01002).

Again the mediating factor was experience. Thus, even a new project manager may not have as much autonomy as one of the more seasoned team members working on the project (B01002). So in this particular team members felt that autonomy overall falls into the moderate range, with more experienced members having a higher level, while new members possessed a lower level.

Feedback. According to one member the amount of feedback and who the feedback comes from was “huge”. This was very important as it related to requirements determination. As in the case in software development projects it was considered of utmost importance to understand the purpose of the application and what the user needed, thus having the “the right feedback from the right people can definitely affect the overall team” in terms of success and moral “. . . you don’t want to be constantly beaten up for something that, you know, you aren’t getting good information about” (B01002). Similarly another member commented, “well for me personal I take the feedback from the customer or the business over the feedback from peers. I mean that is who we are ultimately trying to please” (B01003).

The importance of good feedback between the onshore and offshore members of the team not just between the project manager and the customer was also cited. For instance if developers are reluctant to give up code until they feel it is correct can slow the development process and lead to a communication gap. In this situation, again it is important to foster an atmosphere of collaboration between the onshore and offshore members so that no on is in fear of being “beat up”. This regular feedback between onshore and offshore is important to keep the project on schedule and to let the project manager know exactly where the developers are at. A daily deliver of code to the repository is one way of keeping this feedback loop intact. One member commented: d,

> It's just so important, I encourage and we are there right now, encourage daily for them to deliver their code, to turn over their code, at least check it in to the repository so that we can see it . . . [B01003] can be reviewing it, seeing if they're, you know, sometime the basic structure of it, the starting point, you know if that is not to the standard or the most easily maintainable as we come back into the code as possible we certainly don't want to let that continue being developed in that way. So we can quickly avert some of those, as we have more feedback and quicker turnaround on seeing things and similarly they might not realize that they have questions that need to be asked (B01002).

For this particular team the primary source of feedback was from the work itself:

> In our situation we don't have feedback other than the code itself as an indication of whether our developers are understanding what is going forward and so forth and I really haven't . . . it is a hard one to answer because it's not an issue unless there is an issue that gets raised (B01002).

With the reliance on actual code to gauge the progress of the development, there were times when verbal communication did not actually reflect the progress. As such it is important to have regular "checkpoints":

> So, having those regular checkpoints which is a constant encouragement as the team, that's one of the . . . a key, vital point to it, is getting that type of feedback, that letting it go, it may not be finished, but we need to see where you are and how you are progressing. (B01002)

With these regular checkpoints , "I am not seeing any detriment to not having that direct feedback with the coders in person" (B01002). Another member emphasized that having feedback from the customer was important for both progress on the project and a sense of meaningfulness when a successful project was delivered.

> If they're excited about, hey, this is going to save me an hour, whatever it is designed to do and in the meantime I hear these guys out in the field and they just make a general comment like, our application is the best application in the company that is a little motivator. Just little things like that are good. (B01003)

Core Norms

With globally distributed team members and the offshore contracting relationship it is important for each team member to understand that they are responsible both to the management of the contracting company as well as their contract employer. With the presence of an onshore lead who is employed by the offshore company it is good to have strong communication channels established to each person knows who to report to, thus:

> Our onshore lead knows that he is accountable not just to me and my manager, but he is also accountable to his management chain and so us being in contact with his superiors and sometimes that is a little bit of a grey area at sometimes (B01002).

Perhaps establishing conduct for communication, conflict, and cultural differences was not so heavily tied to the global nature of the team. It could possibly be a consideration on any team:

> So, I really don't think that should . . . that's just people . . . people's people . . . and it doesn't matter if they're . . . you know if you're misunderstood that is not a

> good thing as far as, I don't know, I guess I wouldn't say that has anything to do with it, being globalized (B01003).

In this case the source of the norm development appears to be the team itself rather than strict organizational procedure. Members echoed similar sentiments that it is more of a matter of personal preference and general dealing of people issues. Consistency throughout the duration of the project was cited as an important issue to be considered (B01002). A good measure of flexibility is provided to the team leadership in how to deal with issues of communication, conflict, and cultural differences. The approach centered "more just the team itself and who is managing the team" (B01002).

Team Composition

Size. Referring to a situation where the team had to be ramped up to meet a large delivery one member alluded to the fact that it placed a heavy burden on the project managers and technical leads in their management of the project and the necessary code reviews (B01003). It would have perhaps been better to avoid large ramp ups within a short time period by keeping the team smaller throughout the year and having good communication, feedback, and review so that projects do not fall behind schedule. The use of a more agile approach has been helpful in avoiding this need for a significant ramp up at the end of the year. As noted

> So, it's been successful for us to keep it as a small team and that level of delivery and what's expected could certainly mean that the team needs to be a few larger or a few less. I would certainly see more challenges in a larger team. So, I would say smaller (B01003).

Another member framed the size issue more in relation to the well-being of a team, rather than to its success. In a larger team there was more opportunity for camaraderie, for the development of personal relationships which might serve to build the team unity. So in relation to team size, this member stated, “So, I think you lose that. As far as the work itself, don’t see a whole lot of difference” (B01002).

Mix. When discussing the mix of the team, one member iterated the need for respect among the team members.

> I think more than anything whether it’s beneficial for similar or dissimilar it is a matter of respect. Knowing that there are differences and people perceive things differently and are driven by different things, it is a matter understanding how important it is to respect those differences when they are there (B01002).

Another member indicated that there must be a balance in the mix of the team. Out of the differences come more opportunities for ideas, creativity, and collaboration. But too many differences may lead to miscommunication or slow down in the progress of the project:

> I think it is both. If you have, if everyone on your team is the same ideas are hard to come by and then again if you have such a diverse team that it is hard to communicate with each other, so I don’t know, a good mix (B01003).

The selection of team members was based upon the criteria of the organization. So, a request is made to the type of need, the technical skills needed, the duration of the project, and so on. In this particular case, the decision was made by the outsourcing company.

Knowledge and Task-Related Skills. Due to the nature of the work new team members need to have adequate enough technical skills so that significant training is not necessary, “we are always giving feedback in those types of areas, but the core is, they have to know how to code and they have to have the technical skills to do that” (B01002). It is very important that they are solid in their technical experience and “that’s why we always request, you know, 3 to 5 years, encouraging for some of the best ones” (B01002).

In some form or fashion the technical lead was in a continuous compensation mode as that individual reviewed code, made decisions about maintainability and structure, and overall application performance. There can definitely be cases where one person compensated for someone else. In such a situation, the one person “just cannot have the throughput of two people and you know you cannot sustain that for any length of time” (B01002).

Interpersonal skills

The consensus of the members was that interpersonal skills are extremely important to the success of the team. Personalities must be taken into consideration, mutual respect must be developed, and clear and regular communication must be established. Issues of personality conflict can definitely serve to halt the progress of the project, cooperation is of utmost importance, “as far as teams and what you have money and time and resources and you can do anything, well you can’t if you don’t have cooperation”. (B01002). Vision for the team was also mentioned in relation to interpersonal skills:

> And that vision to me is very interwoven with keeping interpersonal relationships within your team moving forward and you're almost at times a salesman for what you are trying to accomplish and keeping the team motivated and moving in the direction that you need to go and that's just not possible if you have conflicts in the interpersonal and so forth (B01002).

Table 38 provides a summary of the findings for structure for team B01:

Table 38

B01 Team Structure Summary of Findings

Team Structure	Findings
Task Design	
Meaningfulness	Balance between Importance to the team members and to others in the organization
	Significance to stakeholders
	Benefit to the customer
	If no one is going to use and no one cares about it meaningfulness decreases
Autonomy	Contributes to the "well-being" of the team
	Allow for creativity and input rather than dictating what the team must do

(table continues)

Table 38 *(continued)*.

Team Structure	Findings
	Predicated on experience level: higher for experienced members lower for new members
	Experience is mediating factor
	Overall autonomy should be moderate with consideration of experience level
Feedback	“Huge”
	Tied closely to requirements determination
	Member must have understanding of the purpose of application and user’s needs
	“Right feedback from the right people”
	Customer or business feedback over peer feedback
	Encouraging offshore members not to hold code
	Atmosphere of collaboration
	Daily deliver of code to the repository as feedback
	Facilitates quicker turnaround and early detection of potential issues
	Primary source: the work itself

(table continues)

Table 38 *(continued)*.

Team Structure	Findings
	Onshore members communication with offshore members via onshore lead
	Important that verbal communication reflects actual progress
	Regular checkpoints
Task Design	Used for gauging progress and creating a sense of meaningfulness
Core Norms	Understanding of the chain of command
	Strong communication channels
	Source: team itself
	Good measure of flexibility
	Consistency throughout project
Composition	
Size	Keep team small and emphasize good communication, feedback, and review to avoid necessity of large ramp up toward project deadline
	Agile approach has helped to avoid significant ramp up
	More challenges in larger teams

(table continues)

Table 38 *(continued)*.

Team Structure	Findings
Mix	Size in relation to well-being of a team, more members, more opportunity to build camaraderie
	Respect among members
	Balance
	Differences provide opportunities for ideas, creativity, and collaboration
	Too many differences may lead to miscommunication or project slow downs
	Selection based upon type of need, technical skills, duration of project – for offshore members decision made by offshore contracting company
Knowledge and Task-Related Skills	Adequate technical skills that significant additional training is not needed
	Must have solid technical experience and know how to code
	Technical lead facilitates continuous compensation as code is submitted
	Compensation is possible at times, but one person cannot do the work of two people for an extended length of time

(table continues)

Table 38 *(continued)*.

Team Structure	Findings
Interpersonal Skills	Develop mutual respect
	Establish regular communication
	Extremely important to success of the team
	Personality conflicts can halt progress
	Cooperation is of utmost importance
	Vision for the team

Team Virtualness

Extent (Degree) of Virtualness

In this study team virtualness was defined as the extent of virtualness of the team based upon the characteristics of temporal distribution, boundary spanning, lifecycle, and member roles. The following section provides detailed analysis of the team virtualness dimension and its related sub-dimensions beginning with the extent (degree) of virtualness. Based upon the characteristics of each sub-dimension this team represented a medium level of virtualness as summarized in Table 39:

Table 39

B01 Extent (Degree) of Team Virtualness

Sub-Dimension	Comment
Temporal Distribution	Distributed across the United States and India
	Time zone difference 10.5 hours
	No overlapping work hours
	Email and teleconferencing
Boundary Spanning	Crossed organizational and cultural boundaries
Lifecycle	Contract members on a 1 to 2 year rotation
	Organizational members may have long duration
	Team tries to retain same contract members
	This is an ongoing project team. The application was used on a daily basis and will always be in need of support.
Member Roles	Organizational and contract members may play multiple roles within the same team
	Organizational members do not play roles in multiple teams
	Contract members may play roles in multiple teams

An examination of the sub-dimensions of virtualness indicated that although B01 was only geographical distributed over two countries there was a significant time zone difference of 10.5 hours. Due to this difference there were no overlapping work hours whereby communication could take place synchronously unless members agreed to meet during odd hours. B01 utilized a limited amount of ICT to facilitate communication.

In terms of boundary spanning the team crossed organizational boundaries in that the members were from two separate organizations. Some members were from Organization B in the United States and others were employees of the contracting company in India. Because the team was geographically distributed across two different countries several cultural boundaries were identified by the members.

In regard to lifecycle the members preferred a longer duration. The organizational employees were stable having been on the team for 3 years or more. The onshore lead typically rotates off every 1 to 2 years, but that still represents a longer lifecycle. The offshore contractors may rotate off the team at any given time, but B01 tried to keep the same offshore members as long as possible.

Finally, in relation to member roles, the organizational employees were open to playing multiple roles on the team when necessary and did not feel that it negatively influenced the team configuration. However, roles were typically well-defined. The offshore members were generally assigned to one of two groups: the mainframe group or the Java group. Offshore members primarily work in the role of developer. Organizational employees do not play roles in multiple teams; however, contract members may be allocated across multiple teams.

Temporal Distribution

Benefits. The flexibility afforded by a global team was identified as one of the benefits of distribution, “certainly the flexibility we have in an offshore team to add members to ramp up and ramp down as we need based on what we're developing at that time is a benefit" (B01002). The capability to hand off work at the end of the onshore work day to the offshore members and having that work completed by the next day when the onshore team returns was also cited as beneficial. So there was definitely the benefit of quick turnaround on certain aspects of the project (B01002).

Challenges. In this particular team when there was turnover with the onshore lead, since that person was the main communication channel between the onshore and offshore teams that presented a specific challenge, especially if it happened during a large delivery (B01002). Related to the use of an onshore lead as the primary communication channel was the challenge of communicating with offshore team members, "so, with that there were definitely challenges of not having that direct communication with our offshore team" (B01002).

The lack of face-to-face communication also posed a continual challenge:

> And I empathize with them knowing that they want to deliver a quality product, but not having that face-to-face and not having that one-on-one communication there was definitely challenging for us to understand and see where they were when there are issues that arise (B01002).

Although the distribution allowed for much more flexibility there were times when the temporal distribution led to delays. This was also related to the dimension of team structure:

> I in some cases specifically this week where a change comes in requirements we can't definitely commit to the customer whether we can adapt to that particular

change by the expected release date or not in that, as I mentioned earlier, we have our offshore team finishing up one item, testing it, unit testing and so forth, sending it for our review and getting feedback and such and moving right along into other items for other releases. In doing that some changes do not make sense to send back to offshore to do (B01002).

See Table 40 for summary of temporal distribution benefits and challenges:

Table 40

B01 Benefits and Challenges of Temporal Distribution

Benefits	Challenges
Flexibility to ramp up/ramp down offshore resources	Turnover with the onshore lead who is the main source of communication with offshore members
Handing off work to the offshore members at the end of the onshore work day	Communication with offshore team is primary handled through onshore lead, no direction communication with offshore members
	Lack of face-to-face communication to help with understanding and issues
	Delay in project due to inability to commit to request by customer until confirmed that offshore members can complete the work

Use of Information and Communication Technologies. Email between members and teleconferencing between onshore lead and offshore developers

Boundary Spanning

Functional. No specific functional boundaries were identified

Organizational. Offshore members are accountable to both contracting company management and onshore organization management

Cultural. As a culture boundary holidays were identified. Comments included: “on cultural, of course there are the simple things of just different holidays and things like that to manage and plan around” (B01002) and:

> Well, I know too . . . we also have different holidays where we had an instances last month we had a deadline and we were informed . . . that is a three day holiday in India, I guess we will wait on this one. We do have that kind of thing (B01003).

Lifecycle

The goal was to try and maintain consistency with the one offshore mainframe team members and the offshore Java team members. Turnover in team members often led to a slow down since the members needed to get acclimated to the new application and the team. The downside was that the offshore resources can change with little notice and the team may not end the year with the same members it started with, as mentioned by the following member:

> Our offshore team truly can vary. We do not now currently have the two, the Java and mainframe resources we started the year with and that is unfortunate to us because we do prefer to have like I said at least one team member on either side that is consistent, we just have seen a lot of benefits there (B01002).

This same member continued by commenting that In addition to changes within the offshore development members the onshore lead may change as well:

> As far as onshore lead, like I said they typically rotate every two to three years. Now that can also vary between you have . . . last year, the last two years actually we had only 50 percent of a particular onshore lead where he was shared with another much large team for his other 50 percent of communication and management in that area. So that happens (B01002).

In addition, the duration that members remain on the same time was dependent upon the budget and financial constraints:

> Here within just the last two months we have gotten a new onshore lead and he is fulltime us, for right now. We don’t know, you know, quarter to quarter how that may change. How budget, analysts months, you know, drive who we can keep and for how long (B01002).

So, from this team’s perspective there was a preference for a longer team lifecycle to Build consistency into the project and to avoid having to ramp up new members on a regular basis:

> The flip-side of that, staying in one spot, I think the longer you are in one area, my personal experience is that you can contribute more, just because I just know from all the experience for how ever many years I have been on this team, there's rarely something that is a surprise. If the users ask you to do something, do something new or just to enhance something I pretty much know what's involved (B01002).

Member Roles

Multiple Roles in One Team. There were some benefits identified in regard to members playing more than one role within a team. As was the case with the technical lead, that person also worked as a developer which provided flexibility because changes can be made without having to send those requested changes back to the offshore team. In addition, in a situation where there were complex changes, the technical lead could "structure out and develop the structure of it and then turn it over to offshore versus kind of sending it to them and letting them flounder" (B01002). This was also the case for the onshore lead, who also served as a developer. From a negative standpoint, overload on one particular member may occur:

> What we can see happening as far as a negative some time is overloading either one of those resources too much although they love/ development it really puts a strain and can impact other areas if they're having to work quite a bit of overtime and so forth (B01002).

Roles Within Multiple Teams

One member primarily agreed that team members playing multiple roles in multiple teams was a challenge and could represent a hindrance, “just because, you know, many times he (onshore lead) would work overtime to handle things. Because, really, his position with our team demanded fulltime, but he also had the position on the other team that demanded whatever it demanded” (B01003). Another team member recounted their personal experience of crossing teams:

> I have experienced that myself negatively. There is another system that is actually owned by another team and my role is almost customer liaison for that system, got to be quite burdensome as far as allowing me to do project management for this team effectively (B01002).

A similar situation was experienced by another member:

> I have done that myself and it’s really hard, where I . . . even been dedicated completely I was employed on the team that I am on now, but I was loaned out to another team fulltime and our team suffered and I was basically would have to come back and help our team get back on track. And it just, it probably helped the team that I went to, but it hurt the team that I left (B01003).

Much of the consideration was the amount of time that must be allocated. Allocation appeared to be a major issue in this team in terms of conflicts related to project delivery and responsibilities, “of course you get into conflicts also of delivery schedules and expectations and so forth. Also, not knowing what another team maybe requiring on an individual can be of impact” (B01002). Even though a person may be allocated 50-50, “.

. . it’s really who gets to them first you know. I am trying to get things delivered so I use them up as much as they allow me to use them” (B01002).

There may be situations, however, when crossing teams can be a benefit as suggested by this member:

> We had one of our offshore people go out sick for about two weeks and really struggling we were able to pick up someone who was actually assigned to another team, but had worked on this before came in with their level of experience they were able to help us save our last release by giving that support across teams for that two weeks (B01002).

Table 41 provides a summary of the findings for virtualness of team B01:

Table 41

B01 Team Virtualness Summary of Findings

Team Virtualness	Findings
Temporal Distribution	
Time Zone	Distributed across the United States and India
	10.5 hours which provides no overlapping office hours between locations
Use of ICT	Email and teleconferencing

(table continues)

Table 41 *(continued)*.

Team Virtualness	Findings
Boundary Spanning	
Functional	No specific functional boundaries were identified
Organizational	Offshore members are accountable to both contracting company management and onshore organization management
Cultural	Differences in holidays
Lifecycle	
Short	N/A
Long	Preference for a longer lifecycle to build consistency and avoid ramping up with new members on a regular basis
	Turnover in offshore members led to slow down in project
	Offshore resources can change with little notice
	May not end the year with the same members on the team that you started with
	Changing of the onshore lead, two to three year rotation
	Allocation of onshore lead often 50%/50% basis and will always be in need of support.

(table continues)

Table 41 *(continued)*.

Team Virtualness	Findings
	Duration of members dependent upon budget and financial constraints
	This is an ongoing project. The application was used on a daily
Member Roles	
Roles Within One Team	Onshore technical lead able to do development which increases flexibility
	Onshore lead also serves as developer so small changes can be made without having to send them to the offshore members
	Must be careful not to overload resources
Roles Within Multiple Teams	Onshore leads who are allocated to more than one team often have to work overtime which can be burdensome
	Dependent upon amount of time that must be allocated
	Can create conflicts related to delivery schedules and expectations
	Creates issue of who gets the member first
	Crossing teams can be beneficial at times when a member is out sick or on vacation, but as a general rule was considered to have more of a negative impact

Team Agility

Extent (Degree) of Agility

For this study team agility was defined as the extent to which the team aligns with the general values and principles of agile methods as stated in the Agile Manifesto as well as with the practices of a specific method. Overall, members were not familiar the Agile Manifesto itself, but were fairly familiar with the general values and principles associated with agile. Both values and principles were adhered to more indirectly and informally. The team had not adopted a specific agile methodology but had implemented several practices that could be interpreted as agile. Table 42 briefly summarizes the team agility dimension:

Table 42

B01 Summary of Team Agility

Sub-Dimension	Comment
Values	Members indicated that the team strives to adhere to the values of the Agile Manifesto more indirectly except for Value 3.
Principles	Overall the team felt that they strove to adhere to the principles of the Agile Manifesto, but did not realize that the principles were summarized in this particular form.
Practices	The team has not adopted a specific agile methodology. In general the team identified practices commonly considered as agile.

As indicated by Table 42 no comment was made in reference to value 1. Although organizationally the documentation process was still waterfall based, the team was able to compensate for this by backfilling. The team believed that values 3 and 4 were adhered to strongly. In terms of the principles of the Agile Manifesto the team clearly followed 8 of the 12 principles. There was uncertainty on two of the principles and one of the principles, number six, was not implemented. Although the team did not specifically use the terminology typically associated with agile practices, the concepts that emerged in the interview process could be interpreted as short iterations, iteration demos, iteration planning, on-site customer, small releases. Based upon an evaluation of these sub-dimensions of agility this team represented a low degree of agility.

Values

This team indicated that the values of the Agile Manifesto were primarily followed indirectly, except for Value 3, “Customer collaboration over contract negotiation”. Table 43 summarizes the specific comments in regard to the adherence to specific values of the Agile Manifesto:

Table 43

B01 Extent (Degree) of Agility for Values

Value	Yes	No	Unsure	Extent of Agility
Value 1: Individuals and interactions over processes and tools				No comment
Value 2: Working software over comprehensive documentation	X			Organizationally the documentation process still followed a waterfall approach due to audit requirements, thus there was the need to backfill to adhere to a agile (B01002)
Value 3: Customer collaboration over contract negotiation	X			Customer was an on-site, internal customer so there was direct access to the customer on a regular basis (B01002)
Value 4: Responding to change over following a plan	X			Team did respond to changing requirements by altering the plan as often as needed to satisfy the customer (B01002)

Principles

In regard to the principles of the Agile Manifesto, one member indicated, “well, you know the . . . we, before the phrase agile was . . . existed or . . . we as a team had been developing and deploying our code according to these principles, the same principles you sent me" (B01003). Principle 1 was considered a "high priority" for the team, as one member commented, "satisfying our customer and getting really beneficial tools into their hands is the whole drive of the project" (B01002). The team indicated that it adhered to Principle 2. In terms of Principle 3, the team worked with iterations varying between three weeks and six to eight weeks dependent upon the requirements. As one member indicated, there are times when short iterations are feasible and "there are other times that just because of the time of change or the, you know, the type of work we are doing, doesn't allow it" (B01003). In regard to Principle 4, the team had very close communication with the customer since the customer was located on-site. Team had the advantage of presenting the product in person and getting immediate feedback and agreement (B01002). It was agreed that Principle 5 was followed. Principle 6 was not followed due to the globally distributed nature of the team. The primary means of communication with the offshore members was facilitated by the onshore lead via email or nightly phone calls. In general the onshore organizational members did not have direct contact with the offshore members. It was uncertain whether the team truly adhered to Principle 7. Following Principle 8 had increased over the last quarter, but was a challenge. Principle 9 was a definite yes. No comment was given on Principle 10. In reference to Principle 11, the team considered itself to be self-organizing. Finally, following Principle 12, the team did have regular reviews. Table 44 summarizes the adherence to the principles of the Agile Manifesto:

Table 44

B01 Extent (Degree) of Agility for Principles

Principle	Yes	No	Unsure
Principle 1: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	X		
Principle 2: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage	X		
Principle 3: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale	X		
Principle 4: Business people and developers must work together daily throughout the project	X		
Principle 5: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	X		
Principle 6: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation	X		

(table continues)

Table 44 *(continued)*.

Principle	Yes	No	Unsure
Principle 7: Working software is the primary measure of progress			X
Principle 8: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely	X		
Principle 9: Continuous attention to technical excellence and good design enhances agility	X		
Principle 10: Simplicity – the art of maximizing the amount of work not done – is essential			
Principle 11: The best architectures, requirements, and designs emerge from self-organizing teams	X		
Principle 12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly			X

Practices

Due to the fact that the team had not adopted a specific agile methodology such as eXtreme Programming or Scrum the common agile-based terminology did not appear during the interview process in terms of specific practices. Table 45 summarizes the utilization of these practices:

Table 45

B01 Extent (Degree) of Agility for Practices

Practice	Description
Short iterations	Three week to three month (B01002)
Iteration demos	Demonstrating the application to the customer to gain feedback (B01002)
Iteration planning	Regular meetings three times a week initially, dropping down to two meetings currently (B01002)
On-site customer	Internal customer facilitating regular communication (B01002)
Small releases	Look for areas of quick delivery and “low hanging fruit” as it were (B01002)

Methodology

Implementation. The adoption of agile has been an overall benefit. The desire to facilitate more customer collaboration and ownership “is really the whole reason we started going down that path last year in all of our demoing and modeling and so forth” (B01002). In terms of how agile has influenced the team one member stated that we have seen “very complex deliveries be successful where they have not been successful in the past, complex concepts and really changing of business . . . not just processes, but a way of thinking about the processes and their data" (B01002). This same member commented:

> It was not until we started using agile development and really getting that collaboration and quick turnaround and getting things out in front of the customer where we ever got the ownership and the accountability with our customer (B01002).

In this members estimation the use of a waterfall approach for such complex problems did not make good sense "because the customer is so much more out of touch until the very end and they get something and it has not been accepted in the past and even reject multi-billion dollar projects because of it" (B01002).

Benefits. In respect to customers agile has been beneficial in that "fully engaged customers [means] better requirements because they are completely involved in the process, they own the requirements that they provide there is such a collaboration in them" (B01002). The short iterations and demos have created a sense of real ownership of both the process and application. Agile has allowed the team to respond much more quickly to changing requirements and has helped the customer to better understand "that sometime that requires rework and things" (B01002). The team has also experienced benefits from their side. Agile has facilitated the development of "more manageable pieces that we can delivery and get out of our way so we can focus on other items has been a benefit to us" (B01002).

Challenges. Because of the geographic distance and the time zone difference between the U.S. portion of the team and the India portion committing to changes in requirements was cited as a challenge:

> We can't definitely commit to the customer whether we can adapt to that particular change by the expected release date or not in that, as I mentioned earlier, we have our offshore team finishing up one item, testing it, unit testing and so forth, sending it for our review and getting feedback and such and moving right along into other items for other releases (B01002).

For some changes it does not always make good sense to send it back to the offshore members. Having a developer onsite helps to alleviate this issue if the change can be done easily. One member also cited the traditional challenges associated with global teams such as no face-to-face communications and no overlapping work hours. The time difference was also considered a more significant challenge due to the fact that "their night is our day and our day is their night. So we are never in the office at the same time, if we are we either come in real early or stay late or they do the same" (B01002). Meeting the internal documentation requirements was also mentioned as a challenge because the process still follows a waterfall approach, "so it is a challenge . . . until our documentation is adapted to this methodology that has been a challenge" (B01002). A final challenge cited was the situation when the contracting company changes offshore members without informing the U.S. members, "so that is a big challenge. It seems like we are constantly training new guy . . . " (B01003).

See Table 46 for summary of issues related to implementation of agile methodology:

Table 46

B01 Agile Methodology Implementation Summary

Issue	Comments
Implementation	Overall benefit
	Facilitating more customer collaboration and ownership impetus
	Agile allowed for complex deliveries than possible with waterfall
	Has created a change in business processes and way of thinking
	Allowing more collaboration and quicker turnaround
	Increased customer ownership and accountability
	Keeps customers better in touch than waterfall approach
Benefits	Fully engaged customers
	Short iterations and demos have created a real sense of ownership of process and application
	Team can respond more quickly to changing requirements
	Allows for more manageable pieces
Challenges	Challenge committing to requirement changes until verified with offshore members
	No face-to-face communication
	No overlapping hours due to time zone difference
	Internal documentation still follows a waterfall approach

Team B02 Description

The following section provides basic information about Team B02 including: size, how members were selected, locations, type of project, how long the team has been in existence, how long the team will be together, familiarity with the Agile Manifesto, which agile methodology and practices have been implemented, what type of agile training the team has received, other projects the team has worked together on, and the success rating of the team. Table 47 summarizes size, selection, locations, and project of team description:

Table 47

B02 Team Size, Selection, Locations, Project, and Duration

Size	Selection	Locations	Project	Duration
10	Need, skill set	One city in the United States and One city in India	Payroll system in a transportation-related industry	Long-term, will continue for foreseeable future

This was an ongoing project team which will continue indefinitely. The existing organizational employees have been very stable while the contractors have fluctuated based upon the size and complexity of the project. Contractors typically were typically on a one to two year rotation. The team does try to retain the same contractors through bridging them through each year with work. One member indicated that a formula was used to determine the number of onshore leads assigned by the contracting company which basically said, one onshore lead can support five offshore developers. All four of the organizational (onshore) members participated in the study. Members from the contracting company were not accessible for the study due to difficulty of obtaining non-disclosure agreements. Table 48 summarizes demographic information about each member who participated in the study while Table 49 summarizes the title, location, time zone, status on the team, and study participation.

Table 48

B02 Demographics for Members Who Participated in the Study

Title	Location	Employment Duration	Present Position Duration	Team Member Duration	Allocation to the team
Consulting System Developer	USCity1	9-16 years	5-8 years	1-2 years	Main
Consulting System Developer	USCity1	9-16 years	9-16 years	3 or more years	Main
Consulting System Developer	USCity1	9-16 years	9-16 years	3 or more years	Main

Table 49

B02 Current Members and Geographic Locations

Title	Location	Time Difference	Current	Study
Manager	USCity1	GMT -06:00 / CST	X	X
Consulting System Developer	USCity1	GMT -6:00 / CST	X	X
Consulting System Developer	USCity1	GMT -6:00 / CST	X	X
Consulting System Developer	USCity1	GMT -6:00 / CST	X	X
Contractor(s) IndiaCity1		GMT -05:30		

Note: GMT = Greenwich Mean Time; CST = Central Standard Time (US); 10.5 hour difference.

The team has not implemented a specific agile methodology, but follows the agile values, principles, and practices in a more general, high-level approach. Considering that the team had not adopted a specific agile methodology terminology commonly associated agile-based practices did not appear specifically during the interview process. However, in speaking with the members, several concepts associated with agile practices which could be interpreted as short iterations and on-site customer. The team had an overall understanding of agile methodologies. In general the team was not specifically familiar with the Agile Manifesto, but had a general understanding of the values and principles. One member commented that they had been following the values without knowing they were stated as such in the Agile Manifesto. In terms of formal training the members indicated that they received little to no training and no other projects, agile or otherwise were mentioned.

B02 Within-Case Analysis

The following section provides a description of the team and analyzes the interview data to explore the response of members in regard to the three dimensions of the theoretical framework: structure, virtualness, and agility. The analysis of each sub-dimension concludes with a summary of the potentially significant findings.

Team Structure

Task Design

Meaningfulness. Criticality was cited as one aspect which heightens the meaningfulness of a project and fosters motivation and a desire to work harder among members, “we do know that we are important and that keeps us on track a lot of time” (B02002). Another member suggested that perhaps age was indication of how hard a member would work based upon how important they thought the project was to the organization, “if I had somebody that was under thirty it would definitely influence how they work. If I had somebody that was over forty it really would not matter. That has been my experience anyway” (B02003). In this members experience older workers can be given the most menial task and do it without question as opposed to “to somebody out of college and it’s like pulling nails to get it down, because in their mind it’s just not glorified enough to get them that dream job, you know” (B02003). Of course the member did not suggest that this was the case 100% of the time, but that typically young developers “are looking for that project or that volunteer that’s going to get them in the limelight and get them the promotion” (B02003).

Autonomy. One member indicated that they were unsure of how much a sense of autonomy influences the team. They suggest that so much depends on the particular management style employed. In this particular team, “at the project management level where we are managing individual projects we pretty much are allowed our own freedom to do it however we think we need to get things done” (B02002).

One member indicated that “from a team lead perspective it very much plays into the success” (B02003). If the lead has the sense that they have the ability to make decisions about the project then they will whatever it takes to help it succeed. Alternatively, if they do not sense they have a measure of control their willingness to facilitate success declines. Thus, “most team leads will do anything to facilitate the success of a project if they know they have become responsible for the projects success – buy in to it, taking the project on as your own” (B02003). This autonomy can be undermined by a micro managing manager who does not entirely support the approach taken by the lead.

The response in regard to the appropriate level of autonomy was somewhat mixed ranging from high (B02002) to moderate low based upon the individual member. One member suggested:

> The level should be at the level that input and suggestions from the team members are allowed and implemented if it helps the overall design and functionality of the project. After all that is the goal of any project, the best product in the appropriate time line for the customer (B02003).

Ultimately, someone must be “in control of the overall direction and design of the project” (B02003).

Feedback. One member stated that feedback was “really important. It’s a very important piece, I think, for everybody” (B02001). In a sense all members desire to be recognized for the work they are doing. Similarly, feedback from the customer was also considered an important element. Members should also be able to provide feedback as well. Another member stated:

> The team members should feel free to contribute and know that if their idea is a better one and the project scope can accommodate it, then it should be adopted. If this doesn’t happen then the lead begins to work in a vacuum (B02003).

As indicated previously, the members felt it was very important for feedback to come from multiple sources such as the project itself and the project manager. However, special emphasis was given for members receiving feedback from the actual customer because that can be a “tremendous boost to them” (B02001). As one member commented, “we’ve been in meetings where the customer will compliment them and when they walk out of there, that is all they are talking about, you know, we did this right, or we did this right” (B02001). Verbal as well as written acknowledgements can be highly motivating to the members of the team. As one member stated in terms of the source of feedback:

> Any feedback should be considered, no one person knows it all. Sometimes upper management has a different eye then those in the trenches doing the work. If the project has political implications to it then it is best to listen to upper management in some cases (B02003).

Core Norms

As one member suggested a set of core norms “should be the first agreement signed when the team is formed. So everyone knows what is expected in these areas” (B02003). This team was representative of a more pure outsourcing relationship where the onshore lead and the offshore developers were actually employees of another organization located in India. As a part of their training they attended a course developed by their organization pertaining to American culture and covering such topics as sports, movies, literature, food, and literature. One of the issues encountered with offshore members coming to the U.S. part of the team as onshore leads was hygiene. This was representative of a cultural norm between the two countries. After speaking with the members involved and explaining the situation it was no longer an issue. So, norms of how to treat others with respect and open communication were illustrated by this particular example (B02001).

Team Composition

Size. In general members suggested that team size should primarily be based upon the size of the project and/or the number of projects supported by the team, “however many people you need to get the job done is the ideal size” (B02002). However, as one member indicated, as teams grow larger there may be a tendency for them to become cumbersome “you really only need as many people as what it’s going to take to get the job done” (B02002).

In this particular team the number of core members was very stable while the number of offshore contractors fluctuated based upon the size and complexity of the project, “the only reason we would bring contractors in is if we had work for them to do on a specific project” (B02001). In the event when larger teams were formed the team was typically broken down into smaller sub-teams with a lead overseeing each of these sub-teams. So in essence the larger team was functioning as several smaller teams (B02003). In this organization the majority of application development teams range in size between five and six members. One specific issue raised with the use of offshore contractors was related to trying to keep those members who are already familiar with the team and its projects and applications, “you would like to keep those around as much as you can” (B02001).

As one member indicated, the number of onshore leads and offshore development members was based upon a formula:

> They’ve got a formula that they look at for that and basically it is, I think, one onshore can support five offshore developers. And it kind of depends on how big the project, again it kind of goes back to the budget and how many analysts months or how much money we have this year, how big is the project we are working on type of deal (B02001).

As a part of this procedure each quarter a statement of work is created and the project manager will inform upper management of what they will be doing over the next three months and how many people they think they will need from the contracting company:

> And they have done this for a lot of years and that is where they have come up with that 1 to 5, and it is kind of a guideline, if we get six guys offshore, they might let us slide a little bit there (B02001)

Mix. One member considered having a diverse team as a “big plus” (B02001). Another member commented:

> Well, I think you probably suffer more from too many similar than you do from too many different, as long as the job skills are there to do the work, the advantage to the dissimilar again comes out with just kicking around ideas and coming up with new ways to approach things (B02002)

The advantage cited in regard to diversity of experience and background was “just having different points of views, different thought patterns as far as how things go” (B02001). Offshore contractors also contributed to this because “they can bring different ideas and different thought patterns as far as how things could be done and how to crack that nut kind of thing” (B02001). Although one member indicated that they had not “seen a big difference among diversity of backgrounds on teams” they did notice that U.S. members and Indian members did “approach the problems from a different angle” (B02003).

In terms of selection of members with the offshore members the contracting company assigns those members based upon the request of the onshore organization. This was true for both developers and for the onshore lead. Thus far this arrangement has been successful and one member said, “we’ve never been really disappointed with anyone that have brought us with” (B02002). The onshore members were selected through the standard organizational hiring procedure based upon the individual’s resume and the need of the team.

Knowledge and Task Related Skills. In terms of knowledge and task-related skills one member suggested that although appropriate skills are definitely necessary, “I don’t think it has to be . . . everyone has to be of a high-caliber. I think the key there is exploiting each member’s best trait and letting each other learn from their fellow team members” (B02003). This was echoed by another member who indicated that a broad range of skills are necessary, not just technical. Project management and organizational skills were identified as critical to the team, thus the team can “take advantage of each others strengths and weaknesses” (B02002). Another issue mentioned was that of knowledge transfer and capturing the expertise of members on the team in the event they leave for some reason, “trying to mine that knowledge from their brain is really important” (B02001).

As far as compensating for members on the team one member indicated, “well, we do try to take advantage of each other’s strengths” (B02002). Due to the ever-changing nature of technology it was considered important for members to help each other when possible and for mentoring and teaching relationships to be developed (B02002; B02003). Ultimately, “if someone’s not as good as one thing or another then you work around that as best you can” (B02002).

Interpersonal Skills. The overall feeling of the members was that although interpersonal skills are important, the team can still work together successfully on a project if strong personal relationships are not established. As one member stated, “okay, it is important, but it’s not going to sink your team, I guess. It makes the work environment a whole lot more pleasant. But it’s not like something cannot get done if they are not getting along” (B02001). Another member commented, “I don’t know that it is absolutely necessary, but, yeah, I think it surely helps” (B02002). Based upon past experience this member recalled:

> I have been on teams where I had team members that didn’t like each other, although, work still gets done, it is one of those things you work around. As long as they’re not butting heads in public or what not you work around it (B02002).

Finally, another member cited that in a situation where members do not get along or mix well, it was important for members to remain professional. In relation to members getting along personally this member commented:

> It’s good if they do. But, that’s not always the case, unfortunately. I think it is very important, you know, but must some people don’t mix, that is just the way it is. And what I try to do is keep it at a professional level, you know, you don’t have to like everyone, but you should be willing to work with everyone (B02003).

Table 50 provides a summary of the findings for structure for team B02:

Table 50

B02 Team Structure Summary of Findings

Team Structure	Findings
Task Design	
Meaningfulness	Criticality of the project
	May be dependent upon age of the member
	May be dependent upon experience level of the member
	Younger developers eager to get in the limelight for promotions
Autonomy	May be dependent upon management style of project management team
	Very important from the perspective of the project lead
	Micromanaging can undermine the project
	High to moderate
Feedback	Members want to be recognized for their work
	Customer and member feedback very important
	Work is not done in a vacuum
	Multiple sources
	Special emphasis on customer feedback
	Verbal and written

(table continues)

Table 50 *(continued)*.

Team Structure	Findings
Core Norms	Important to be agreed upon at start-up to define expectations
	Hygiene issue: example of open communication and respect
Composition	
Size	Based upon size of project and number of projects supported by the team
	Larger teams can become cumbersome
	Larger teams can be broken into smaller sub-teams with a project manager for each
	When offshore contract members are employed try to keep these members
	Establish a formula for onshore leads and offshore developers
	Budget consideration / Statement of work
Mix	Kicking ideas around and coming up with new approaches
	Good mix provides different points of views
	Contract members bring in experience from working with other organizations
	Offshore members selected by contracting company based upon organizational request of need skill set

(table continues)

Table 50 *(continued)*.

Team Structure	Findings
	Organizational members selected based upon standard hiring procedure
Composition	
Knowledge and Task-Related Skills	Members do not have to have expertise in all areas of the project
	Key is exploiting each member’s best traits and strengths and then allowing them to learn from each other
	Broad range of skills necessary, not just technical
	Project management and organizational skills are crucial
	Development of strategies for knowledge transfer and capturing expertise of members
	Compensation is necessary and acceptable
	Development of mentoring and teaching relationships
Interpersonal Skills	Important, but the team can still work together and complete a project even if strong interpersonal relationships are not established
	Even when members do not get along personally, they should still retain their professionalism

Team Virtualness

Extent (Degree) of Virtualness

In this study team virtualness was defined as the extent of virtualness of the team based upon the characteristics of temporal distribution, boundary spanning, lifecycle, and member roles. The following section provides detailed analysis of the team virtualness dimension and its related sub-dimensions beginning with the extent (degree) of virtualness. Based upon the characteristics of each sub-dimension this team represented a medium level of virtualness as summarized in Table 51:

Table 51

B02 Extent (Degree) of Team Virtualness

Sub-Dimension	Comment
Temporal Distribution	Distributed across the United States and India
	Time zone difference 10.5 hours - no overlapping hours
	ICT use included teleconferences, Microsoft® Visio, email
Boundary Spanning	Crossed organizational and cultural boundaries
Lifecycle	Offshore members on a 1 to 2 year rotation; length of stay also impacted by funding and need
	Onshore members may have long duration
	Team tries to retain the same offshore members
	Ongoing project team

(table continues)

Table 51 *(continued)*.

Sub-Dimension	Comment
Temporal Distribution	Distributed across the United States and India
	Time zone difference 10.5 hours - no overlapping hours
	ICT use included teleconferences, Microsoft® Visio, email
Boundary Spanning	Crossed organizational and cultural boundaries
Lifecycle	Offshore members on a 1 to 2 year rotation; length of stay also impacted by funding and need
	Onshore members may have long duration
	Team tries to retain the same offshore members
	Ongoing project team
Member Roles	Organizational and contract members may play multiple roles within the same team
	Organizational members do not play roles in multiple teams
	Contract members may play roles in multiple teams

An examination of the sub-dimensions of virtualness indicated that although B02 was only geographical distributed over two countries there was a significant time zone difference of 10.5 hours. Due to this difference there were no overlapping work hours whereby communication could take place synchronously unless members agreed to meet during odd hours. B02 utilized a limited amount of ICT to facilitate communication.

In terms of boundary spanning the team crossed organizational boundaries in that the members were from two separate organizations. Some members were from Organization B in the United States and others were employees of the contracting company in India. Because the team was geographically distributed across two different countries several cultural boundaries were identified by the members.

I regard to lifecycle the members preferred a longer duration. The organizational employees were stable having been on the team for at least one year. The onshore lead typically rotates off every 1 to 2 years, but that still represents a longer lifecycle. The offshore contractors may rotate off the team at any given time, but B02tried to keep the same offshore members as long as possible.

Finally, in relation to member roles, the organizational employees were open to playing multiple roles on the team when necessary and did not feel that it negatively influenced the team configuration. However, roles were typically well-defined. The offshore members were generally assigned to one of two groups: the mainframe group or the Java group. Offshore members primarily work in the role of developer. Organizational employees do not play roles in multiple teams; however, contract members may be allocated across multiple teams.

Temporal Distribution

Benefits. Having onshore leads from the contracting company who can also program has been a benefit to the team because they can also be working on code during the day and hand that code off to the offshore team at the end of the U.S. work day. As one member indicated:

> That is one of the stronger points of the offshore . . . of offshoring, once you have built the trust and . . . our application is pretty complicated, at least we think it is, so once you have built the trust and the expertise of working on it, it gets pretty easy to hand off pretty big assignments and when it comes back you have a pretty good idea that it is the way it is supposed to be and everything works pretty good there and you generally get stuff back quicker than you had planned (B02002).

Challenges. Communication was cited as a significant challenge "because you're dealing with a different culture, you know, and you really have to understand how that culture communicates . . . " (B02002). One difficulty related to communication and cultural differences identified was that Indian members had a tendency not to speak up when they did not understand something. It was deemed important that the team should foster an environment where members feel comfortable to ask questions. Language was also identified as a communication challenge especially "everyday terms that you and I would use and not think nothing of it kind of throws them for a loop because their learning an English or British version of English and sometimes it is a little confusing" (B02003). A second challenge mentioned was in reference to the large time zone difference, "if something comes up at noon today that is midnight in India and in general you are not going to find somebody" (B02002). Thus, with no overlapping hours in the work day contacting offshore members was cited as difficult when onshore members have an issue that needed immediate attention. See Table 52 for summary of temporal distribution benefits and challenges:

Table 52

B02 Benefits and Challenges of Temporal Distribution

Benefits	Challenges
Onshore lead can also do development	Indian members may not speak up when they don't understand something
Handing off code back and forth after onshore or offshore work day	Communication / Language barrier
	Large time zone difference, no overlapping hours

Use of Information and Communication Technologies. Much of the communication of the team is handled via the onshore lead from the contracting company who serves as a liaison between the onshore and offshore members. However, conference calls are conducted and the offshore contractors are listed in the email system, so onshore members can contact them directly when necessary. One member indicated that conference calls are often more beneficial because it was "easier to get a read as far as how well they understand something or if they're asking the right questions or maybe the tone of their voice when they are asking those questions, that type of thing" (B02001). The team has not implemented chat technologies to avoid tying up communication bandwidth. In addition, "visual" communication was facilitated on occasion via Microsoft® Visio documents, “If I can create a Visio document or flowchart of what it is that I want them to understand I will do that and that seems to help quite a bit” (B02003). Versioning tools were used to keep track of changes.

Boundary Spanning

Functional. No specific functional boundaries were identified.

Organizational. Initially there was some organizational resistance to the idea of offshoring. However, over time members began to see that offshore members were very competent, knew what they were talking about, and were friendly, outgoing people (B02001). Previously, the contractors were from American companies and worked primarily as individuals without particular bond or relationship to one another. This was actually a downside to having domestic contractors. In that situation every time a new contractor joined the team they had to be trained individually and when “ne of them leaves and another comes in from the same company, I am stuck right back where I was, aren’t I, I am training them from scratch, ramping them up from scratch” (B02002). This boundary has actually been alleviating by working with an offshore contracting company because the contracting company provides training for the offshore members, “one comes in and they as a company train and completely get that person ready and ramped before they even bill us” (B02002). Another organizationally related concern was the issue of the seniority based approach of the contracting company. As one member commented, “you know, if you’ve been there longer than anybody else you are going to be the senior person that gets the lead job whether you’re really qualified or not” (B02003). Thus, there may be situations where the junior employee may do a better job as the onshore lead than the senior employee.

Cultural. Common cultural boundaries such as the language differences and mutual understanding were identified. However, over time as the members have worked together and gotten used to differences in accents and word usage.

Lifecycle

In respect to the lifecycle of the team one member stated, “I don’t know that the time matters that much, it just really comes down to the quality of the person that is doing the work” (B02001). Teams within this organization generally have a one to two year rotation of the offshore members. For example, the contracting company will typically work a person offshore for a couple of years and then send that person to work onshore for a year and then rotate them out around the first of the year. This particular team utilized a two-year rotation for offshore members and this member of the team commented, “. . . I don’t know if the time was a big a factor as just the individual” (B02001). Rotation was primarily among the offshore members. Onshore members from the organization have typically stayed for long periods of time. As a rule, the organization did not have any specific set time constraint on how long onshore members stayed on the team, “so a lot of us have been here a long time and we’ve been in our same jobs for a long time” (B02002). One member, however, did indicated that “if it’s a large system that has a lot of maintenance the longer team seems to be better, because it is just gathering more experience and expertise” (B02003). An additional element of the lifecycle of the team within this organization in regard to offshore members was funding as indicated by the following member:

> On the offshore team, you know, it fluctuates based on our funding for the coming year and what we try and do is maintain that same team by bridging them through each year with work until they actually have to leave the team. So the offshore piece, the contractor piece, we try to maintain the same team all the time unless we are forced to release somebody (B02003)

Member Roles

Multiple Roles in One Team. Overall the members considered playing multiple roles within the same team as a benefit as long as the roles are managed responsibly. Sometimes member can have tendency to bite off more than they can chew (B02002). Serving in multiple roles can keep members from becoming bored, make the project more interesting, teach them how to set priorities, and help them to see the value of the roles that each member plays in the team (B02001; B02003).

Roles Within Multiple Teams. The assignment of members to roles on more than one team was considered a challenge to a successful team configuration. In a situation where a member was crossing teams one member stated, "And I would say we drove her crazy or drove her batty" (B02001). However, the ability to function effectively on more than one team may also be related to the individual person:

> Yeah. And again that varies by developer too. If you've got pretty experienced developer you never really notice an issue because they just get the work done.

> If it's a younger developer, you know, he struggles with who he needs to be faithful with and what needs to get done (B02003).

Priorities can also pose an issue. If one team's project has a higher priority over the other team's project it may hurt the team with the lower priority as suggested by this member, "because she was devoting so much time to my project so I think it kind of hurt on the other side, they did not get as much done as what they were wanting to last year" (B02001). As another member commented:

> One person's priority is not the other person's priority and you get into conflicts like that. You end up having a meeting with both parties in the room and saying, "okay, you know, you two business customers come to an understanding of what's priority, because I only have one person and you are tugging him two or three different ways" (B02003).

In addition to offshore developers, onshore leads from the contracting company may also be allocated across more than one team. This has the potential to be both a benefit and a challenge. Some onshore leads are able to do this and others are not, much of it depends on the specific individual. In fact according to one member there was an onshore lead that "complained she doesn't have enough to do" (B02002). Table 53 provides a summary of the findings for virtualness or team B02:

Table 53

B02 Team Virtualness Summary of Findings

Team Virtualness	Findings
Temporal Distribution	
Time Zone	10.5 hours which provides no overlapping office hours between locations
Use of ICT	Communication between onshore and offshore members handled primarily by onshore lead serving as liaison via teleconferences and email
	No chat technologies
	Visual communication via Microsoft® Visio
Boundary Spanning	
Functional	None identified
Organizational	Initial resistance to offshoring
	Seniority based approach to assigning onshore lead
Cultural	Language differences and mutual understanding

(table continues)

Table 53 *(continued)*.

Team Virtualness	Findings
Lifecycle	
Short	N/A
Long	May be more dependent on quality of person rather than time
	Offshore members are on a 1 to 2 year rotation
	Onshore members have stayed for longer periods of time
	Longer lifecycle is more beneficial for large systems with a lot of maintenance
	Offshore members based upon need and funding
	Team tries to retain the same offshore members through bridging them through each year with work
Member Roles	
Roles Within One Team	Beneficial as long as roles are managed responsibly
	May alleviate boredom, keep project more interesting, teach prioritization skills, and instill appreciation for what other members do on the team
Roles Within Multiple Teams	Dependent upon the individual
	May cause conflicts between which team's project takes priority
	Offshore developers and onshore lead may be allocated across multiple teams

Team Agility

Extent (Degree) of Agility

For this study team agility was defined as the extent to which the team aligns with the general values and principles of agile methods as stated in the Agile Manifesto as well as with the practices of a specific method. Overall, members were not familiar the Agile Manifesto itself, but were familiar with the general values and principles associated with agile. Values were adhered to more indirectly and informally. The team had not adopted a formal agile methodology, but had implemented a few practices which could be understood as agile. Table 54 briefly summarizes the team agility dimension:

Table 54

B02 Summary of Team Agility

Sub-Dimension	Comment
Values	Team definitely felt it adhered to Value 3 directly, but the other values were implemented more indirectly
Principles	No data were provided
Practices	The team has not adopted a specific agile methodology. In general the team identified the following practices commonly considered as agile: short iterations and on-site customer

As indicated by Table 54 the team believed that they adhered to the values of the Agile Manifesto in general. However, it was noted that getting around the organizational documentation requirement was a challenge. Because the team was not familiar with the principles of the Agile Manifesto no specific responses were received. Although the team did not specifically use the terminology typically associated with agile practices, the concepts that emerged in the interview process could be interpreted as short iterations and on-site customer. Based upon an evaluation of these sub-dimensions of agility this team represented a low degree of agility.

Values

In response to the question, “were you familiar with the Agile Manifesto?” the following member responded, “I can’t say that I was familiar with it before, but I . . . you sent this at one point . . . you sent that document and I kind of read through it” (B02003). However, although the member was not previously familiar with the Agile Manifesto, he commented, “all those four things I’ve actually done over the years that I have done this” (B02003). Specific comments are summarized in Table 55:

Table 55

B02 Extent (Degree) of Agility for Values

Sub-Dimension	Yes	No	Unsure	Extent of Agility
Value 1: Individuals and interactions over processes and tools	X			Emphasis on interacting directly with members rather than too many formal meetings (B02003)
Value 2: Working software over comprehensive documentation	X			Due the controls surrounding documentation within the organization it was difficult to get around certain documents which must be in place to proceed with the project (B02001, B02003)
Value 3: Customer collaboration over contract negotiation	X			Due to an internal, on-site customer, collaboration was considered very high within the team (B02003)
Value 4: Responding to change over following a plan	X			Flexibility was emphasized and recognition was given to the fact that following a formal process to the letter was difficult (B02003)

Practices

Considering that the team had not adopted a specific agile methodology terminology commonly associated agile-based practices did not appear specifically during the interview process. According to one member the formal development process "for a new project or new development it is pretty much a standard lifecycle" (B02002):

> Course it goes through the customer decides they want something and present it to use in their terms, and we go back and talk over how we might approach it and we work up estimates off of that and feed that back to them and make sure . . . what we can do is what they were wanting, so it is pretty much a lifecycle and once that's gone back and forth a couple of times of course you can start developing and testing and implementing (B02002).

In regard to the process utilized by B02, one member made the following comment:

> My particular version of agile development is really a variation of the formal process. The things that bog us down I don't really use unless I absolutely need to. Like, I said, the overall thing, I have got a palette of 20 tools in the formal process or more, but I pick and choose those tools that I need as appropriate (B02003).

As this member continued, this particular agility process may be:

> . . . as short as a week, sometimes even three days to . . . I mean it can be up to a month before it finally goes into production just depending on how concerned the customer is and wanting to test all the different scenarios (B02003).

Table 56 summarizes the implementation of agile practices:

Table 56

B02 Extent (Degree) of Agility for Practices

Sub-Dimension	Description
Short iterations	One week up to one month (B02003)
On-site customer	Internal, on-site customer facilitating regular communication (B02003)

Methodology

Implementation. In terms of training involved with the implementation of agile the team had some basic project management courses which included a discussion on agile, but it was not specifically focused on it (B02001). Another member cited that they had not had any agile training in particular:

> A few years ago I was involved with some stuff that . . . something similarly called RAD or Rapid Application Development was big buzz word in the 90s. But, we used call it “fail forward fast” which was, you know, our customers had high demands and did not want to wait for deliveries, so we would put stuff up similarly to the agile development. We would make get stuff developed and moved as fast as we could and in the next iteration correct the errors. Getting them closer and closer at each iteration and plus adding new functionality as we went along and we just called that fail forward fast. Meaning you don’t want to fail

> backwards when you make mistakes, you continually move forward. But that's as close as I can come to what you are talking about (B02002).

In response to the question, "were you familiar with the values and principles of the Agile Manifesto before the interview?" one member responded:

> No. Not prior to your passing them on. We do have an agile development team here, they have been trying to get people to pay attention to . . . it is actually not in [Confidential], it is in [United States]. So I have seen the concept is there and that people are trying to have us look at it, but I haven't spent any time doing that (B02002).

Benefits. One benefit of global distribution identified was the ability to ramp up the team when a large development project begins. As one member explained:

> Say we need to bring 13 people on for this project. For us to go out and hire 13 people off the street just won't happen in the time period that we need to get a project done, I mean, we usually do a project on a year to year basis, trying to bring 13 people in and bring them up to speed and get them going with a project is kind of tough. For us to go hire them, bring them in, and do all that (B02001).

Working with the offshore contracting company "they can ramp up people and they've got a real knowledge transfer process where they can bring people up and get them going in a relatively short period of time" (B02001).

The same holds true for ramping down the team when the project is completed. The added benefit is that the onshore project management becomes familiar with the offshore resources so they can be requested again when another large project is scheduled, “the knowledge transfer that they provide, that they use, that is a big benefit” (B02001). Although this is not directly related to the agile method, it does increase the agility of the team in terms of its ability to be ready for large projects and its ability to decrease when smaller projects are assigned.

On the maintenance side another aspect of global distribution that benefited the agility of the team was the capability to send problems at the end of the U.S. work day to the offshore team and they can have it done when we walk in the door in the morning and we can keep going. So there’s some benefit there as far as that time when you have around the clock production” (B02001). The offshore component of the team also benefits overall agility when scheduling meetings with clients. For example, the U.S. team can meet with the client one day to discuss project design . . .

> we can write those changes up, send them to the offshore team, they can come in and change the design, rework the documents, and we can have a meeting the next day with them, the customer, again and go back over the changes that have been made or the revised documentation and make sure everything is good and keep moving. So, there’s . . . that time difference, having people work around the clock basically is a big plus.

Challenges. Just the newness of the agile was cited as a challenge when the team initially began to implement it. One member commented that a key to overcoming the challenges involved with agile was getting the weekly meetings setup. This member stated:

> This was the big piece, the communication piece, to make sure when we are talking to them that they do understand what we are doing and some of the review stuff as far as when the design comes back, the approach document comes back, reviewing that to make sure that what we have communicated to the onshore project lead is understood by the offshore team as well (B02001).

Having this in place ensured that "everything was going to be understood and that communication, that there was open communication there and it has worked out pretty well for us" (B02001). See Table 57 for summary of issues related to implementation of agile methodology:

Table 57

B02 Agile Methodology Implementation Summary

Issue	Comment
Implementation	Had some basic project management training which included a discussion on agile, but it was not specifically focused on it
	One member cited no training
Benefits	Ramping up / ramping down the team quickly
	24x7 coverage, handing off work, round the clock production
Challenges	Newness of agile
	Getting the weekly meeting setup
	Understanding and communication

Organization C Case

Organization C Description

Organization C is a global technology solutions provider based in the United States with approximately 9,000 employees distributed in 45 countries. Organization C has adopted agile methodology from the top down. These methods are supported by the organizational leadership and their use is highly encouraged. Employees are required to attend mandatory training in agile development. Agile processes have begun to cross functional boundaries moving out of the information technology department and into the marketing department as well.

Team C01 Description

The following section provides basic information about Team C01 including: size, how members were selected, locations, type of project, how long the team has been in existence, how long the team will be together, familiarity with the Agile Manifesto, which agile methodology and practices have been implemented, what type of agile training the team has received, other projects the team has worked together on, and the success rating of the team. Table 58 summarizes size, selection, locations, and project of team description:

Table 58

C01 Team Size, Selection, Locations, Project, and Duration

Size	Selection	Locations	Project	Duration
21	No specific criteria was provided	One city in the United States and One city in Poland	Web-based share management tool within the travel industry	This team has been in existence in its current form since 2006 and is an on-going, long-term team

Although the team in Organization C has not implemented one specific agile methodology in its entirety, it has tailoring the practices of eXtreme Programming and Scrum to fit its development needs. Such practices as daily stand-up meetings, iteration planning, user stories, velocity, Test-Driven Development, continuous integration, common code, standards of code, simple design, unit test, automated test, short iterations, acceptance criteria/test, iteration demos, iteration retrospectives, pair programming, and refactoring have been implemented and used effectively.

The team, in its form at the time of the study, had been in existence since 2006. A historical review revealed that the two teams were already in existence separately but due to a need to be more globalized and establish a low cost presence Organization C decided to merge the teams in 2006 to re-architect a specific product. Thus, the members who were on each team respectively became members of a single team. The Poland team was acquired through a buy-out of another organization. The decision to merge the teams was based upon the fact that the United States team was in the process of re-architecting its product and needed additional support, so the Poland team, whose project was just canceled, was brought in to form a new globally distributed agile team. Of the 21 current members eleven members participated in the study. Table 59 summarizes demographic information about each member who participated in the study while Table 60 summarizes the title, location, time zone, status on the team, and study participation:

Table 59

C01 Demographics for Members Who Participated in the Study

Title	Location	Employment Duration	Present Position Duration	Team Member Duration	Allocation to the team
Project Manager	USCity1	5-8 years	3 years or more	3 years or more	Main
Senior Application Systems Analyst	USCity1	9-16 years	5-8 years	3 years or more	Main

(table continues)

Table 59 *(continued)*.

Title	Location	Employment Duration	Present Position Duration	Team Member Duration	Allocation to the team
Supervisor	PolandCity1	5-8 years	6-12 months	1-2 years	Main
Manager	USCity1	17-24 years	3-4 years	3 years or more	Main
Quality Control and Testing Contributor	PolandCity1	3-4 years	3-4 years	1-2 years	Main
Quality Control and Testing Contributor	USCity1	17-24 years	3-4 years	3 years or more	Main
Senior Software Developer	PolandCity1	5-8 years	1-2 years	3 years or more	Main
Senior Software Developer	USCity1	5-8 years	5-8 years	3 years or more	Main
Software Developer/Analyst	PolandCity1	1-2 years	1-2 years	1-2 years	Part
Application Architect	USCity1	9-16 years	3-4 years	1-2 years	Main
Senior Technical Lead	USCity1	9-16 years	5-8 years	3 years or more	Main

Table 60

C01 Current Members and Geographic Locations

Title Location		Time Zone	Current	Study
Project Manager	USCity1	GMT -06:00 / CST	X	X
Senior Application Systems Analyst	USCity1	GMT -06:00 / CST	X	X
Senior Technical Lead	USCity1	GMT -06:00 / CST	X	X
Senior Software Developer	USCity1	GMT -06:00 / CST	X	X
Software Developer	USCity1	GMT -06:00 / CST	X	
Quality Control and Testing Contributor	USCity1	GMT -06:00 / CST	X	X
Quality Control and Testing Contributor	USCity1	GMT -06:00 / CST	X	
Manager USCity1		GMT -06:00 / CST	X	X
Application Architect	USCity1	GMT -06:00 / CST	X	X
Supervisor/Business Analyst	PolandCity1	GMT +01:00	X	X
Quality Control and Testing Contributor	PolandCity1	GMT +01:00	X	X
Quality Control and Testing Contributor	PolandC ity1	GMT +01:00	X	
Business Analyst	PolandCity1	GMT +01:00	X	
Business Analyst	PolandCity1	GMT +01:00	X	
Software Developer	PolandCity1	GMT +01:00	X	X
Software Developer	PolandCity1	GMT +01:00	X	
Software Developer	PolandCity1	GMT +01:00	X	

(table continues)

Table 60 *(continued)*.

Title Location		Time Zone	Current	Study
Senior Software Developer	PolandCity1	GMT +01:00	X	X
Developer PolandC	ity1	GMT +01:00	X	
Developer PolandC	ity1	GMT +01:00	X	
Developer PolandC	ity1	GMT +01:00	X	

The team has not adopted a specific agile methodology but rather has tailored specific practices from eXtreme Programming and Scrum to its specific situation. The team was extremely agile in terms of practices which were implemented. These agile practices included daily stand-up meetings, iteration planning, user stories, velocity, Test-Driven Development, continuous integration, common code, standards of code, simple design, unit test, automated test, short iterations, acceptance criteria/test, iteration demos, iteration retrospectives, pair programming, and refactoring. The members were very familiar with the agile methodology. Although the members were very familiar with the agile methodology and practices commonly associated with it, they were not as familiar with the Agile Manifesto itself. After seeing the Agile Manifesto the members indicated that the team did indeed adhere to many of the values and principles. It was more of a situation where they had not seen the values and principles of agile presented in this particular form. Beginning in 2005 training began to be required of project managers. In 2006 it became mandatory for all employees. Members now go through basic agile classes. So, in 2006 everybody on this team was trained. All new members brought onto the team will be required to go through the mandatory agile training. All of the projects the team has worked on have been agile.

Rating of Team Success

The success of this team has been recognized within the industry by receiving a best in class award from the Forrester group. According to one member, “within [Organization C] we’re recognized as a very successful agile team in which individuals are sought out to go help other teams” (C01001). When asked, “what makes the team successful?” the answer was:

> We have the support of our leadership to do what we need to do and our results prove that we are able to make the right decisions for what we need to do and, you know, you have that autonomy, you know, you have that support, you can be successful in your agile practices (C01001).

Overall the members of this team believed that they have been successful in their agile projects having overcome various initial challenges (C01003; C01006; C01007; C01008; C01009; C01010).

C01 Within-Case Analysis

The following section provides a description of the team and analyzes the interview data to explore the response of members in regard to the three dimensions of the theoretical framework: structure, virtualness, and agility. The analysis of each sub-dimension concludes with a summary of the potentially significant findings.

Team Structure

Task Design

Meaningfulness. One of the important issues revolving around the meaningfulness of the work being done in the project is the sense that it is part of something larger. As one team member stated, “when they do understand that big picture and what they are contributing to it that is helpful as well. I know it is easy sometime to get caught down in the trenches and not see the whole picture (C01004)”. For team C01, meaningfulness is related to multiple concepts: importance, usefulness, quality, accountability, responsibility, understanding, ownership, trust, motivation, growth, and challenge. For this team importance is not necessarily defined by the size or priority of the project itself, but rather by it’s usefulness to the intended user:

> I believe it is always not only . . . it is true for development; people prefer to do something that is important somehow. But it also doesn’t mean that people expect that they . . . the product they are working on has a very high priority or marketing release or something like that. They just need to know that it is useful, that somebody will use it and I think this is important. It don’t need to be a big project just I would say just to know that it is useful, it could be important, but as I said important not in the meaning of high priority or marketing priority (C01003).

Interestingly, in one interview the team member suggested that ownership actually takes precedence over importance and communicated this idea by using the analogy of the parent-child relationship when it comes to describing the significance of meaningfulness to a successful team configuration, “I really think what makes people harder is product ownership. If they feel responsible for this product, even though not very important, they naturally start working harder on it. So, it is like their baby" (C01007).

In relation to the interrelationship between meaningfulness, challenge, trust, accountability, and growth, a team member made the following comment:

> I think if . . . yeah, from my experience, you know, you know you are giving somebody a challenging piece of work, it shows that you trust them and that you are holding them accountable and you know it's always been the way I have seen people grow, you know (C01004).

The connection between meaningfulness, motivation, and quality may also lie in having a clear understanding of the overall goal of the project:

> If you're . . . the better you understand what you are doing the better are the results that you may provide. If you don't understand what you are doing you just do something else and you are not so motivated and you're just focusing on, you know, working eight hours, yeah, so I think this is important factor for motivation if you are understanding the importance of the goal (C01005).

Also related to motivation is working on projects which may be establishing a new path in the organization, such as in this case the move to an agile methodology for converting an existing legacy system, "I don't how to call it, important or . . . we, for example, I believe that me, for sure it was really great motivation for me to work harder " (C01009).

Another aspect of task design and meaningfulness relates to how work is distributed among the team. In this particular team many different roles are present:

architects, developers, business analysts, testers, etc. At the intersection of agility and structure an agile practice such as small iterations can contribute to the meaningfulness of the project:

> So, I think it is just basically because we do it in iterations I think that each small iteration is more . . . is really the important factor because once those iteration are . . . that's successful then the overall project will be successful (C01006).

Continuing with this idea of taking the project from the big picture to the specific details implemented through short iterations is highlighted in the following comment when multiple user requests are made which might make the team feel a little overwhelmed by the project complexity and size, ". . . so, we need to be able to do this, this, this at a high level and then break it down, whittle away at it (C01008). One note of caution was in regard to the importance of the project and decision-making was pointed out by another member who stated,

> I am not sure if it is related to important project, well, maybe yes, people start working harder when the project is important, but in my opinion when the project is important it . . . for some people it may be harder to make a decision, yeah (C01007).

Autonomy. Although the project manager serves as the coordinator of the team and provides guidance for focusing on the "right priorities at the right time" (C01002), there was an overall consensus on the team that autonomy is crucial to the team configuration:

> So there is no, you know, there's no you know . . . our project manager just makes sure that we are, you know, doing everything in a more time efficient manner, but everybody is always . . . every time you have a phone call or NetMeeting everybody is encouraged to give their feedback (C01006).

Multiple comments were made in regard to the influence of autonomy on the team as evidenced by statements made by these three members respectively. The first member stated, "I would say for my team here in [Poland] and I could also answer from this perspective, it is a crucial team [element]" (C01003). The second member concurred, "it is important to [have] autonomy. I can't tell about other teams in [C01] that they work using agile, but in our team it is important to give people autonomy" (C01007). And finally, a third member simply commented, "Okay. I believe that autonomy is a really important thing" (C01009).

Within the priorities set by the project manager, autonomy may come in the form of small groups of members working together to come up with a solution to a requirement or problem:

> Within the priority I would say individuals have a lot of autonomy . . . not necessarily the individual . . . but when you get down to the smaller team of two or three working together they have a lot of autonomy in trying to come up with the best approach. I think that daily call and the relationship that we have facilitates this kind of back and forth, so interdependencies and coordination just . . . can be done fairly quickly, we're all really good at shooting off a quick email, hey, "what do you think about this?" (C01002).

Autonomy was also coupled with prioritization and scope. As one member commented:

> There's not this sense that, "oh, I got to check with the project manager". Unless, it is obvious to people involved that this is going to increase the scope or this is going to impact our estimates or something like that, then, yeah, we tend to go to her and get some direction for the sake of prioritization and if necessary we go back to the market advocate for trade-offs and prioritization purposes (C01002).

Autonomy may also depend on the level of experience and expertise of the team members:

> But I would say fairly . . . I guess when they are new, when we get these new, young kids coming in, there's, you know, a little more oversight, but with some of the senior developers they feel, I would think they feel fairly autonomous and able to make decisions (C01002).

Autonomy was related to practices and tasks:

> It is very important and this was many times a big issue when we had the situation where somebody from outside tried to change our practices, to introduce some practices that are not suitable for our practice or something like that. It was always a big challenge for our team, we would rather . . . to have the possibility to decide how we are working, what practices we are following on this task (C01003).

Autonomy was also linked with the concepts of ownership and accountability. When team members are split across distributed sites there may be a tendency for one of the team's to sense a loss of project ownership (see relationship between ownership and meaningfulness). In this particular team, the marketing people are located only in one site and the majority of business decisions are made at this site. If there is not an explicit effort made to communicate these decisions and why they were made to the other site, ownership may be lost. In an effort to remedy this challenge, a conscious effort has been made to begin to split ownership across sites, as indicated by the following member:

> So in moving forward we have the other projects we're actually splitting the ownership, the ownership of this project is here, so it's not all in one place, so people feel like they are owning more than they did before, which is one . . . like one of the reasons why we are having them handle the implementations because that's a huge part of feeling accountable to . . . is getting that into production and keeping it there, you know and testing it out. So, as we move forward this year we have more of an opportunity to give them additional responsibilities. As I said, our architect had to move on so we split the responsibilities some individuals here and some individuals there, you know, and I think that has helped a lot to feel more ownership and accountability (C01004).

In addition to ownership, the concepts of accountability and shared responsibility were also highlighted, "we have something like PMD checkers, we have unit test, miscellaneous metrics generated after nightly builds, etc., but people in natural way start to feel more responsible when they have more autonomy" (C01007).

Level of autonomy may be separated by its relationship to business decisions as opposed to technical decisions, "within the code we provide, you know, we can do whatever changes as long as the code works and use different tools. There are some exceptions connected with infrastructure, but basically we have some of autonomy"

(C01005). So, in terms of the algorithms that are written and their implementation, this is the “brainchild” of developers and when questions or problems come up it is a matter of making a call to the project management team and saying,

> “hey, we can work this out”, and pretty much . . . and generally the senior people kind of dictate, “hey, you . . . give them a little heads up and say your gonna face some problems here or just be careful here, or just let me know when you are coming to this part and we talk about it” (C01008).

So flexibility in regard to the technical aspect of the project is left to a large degree in the hands of the development portion of the team, as the following team member described:

> when it comes to technically or other things it’s, everybody contributed to create, like if you want to design something or present something to the user, everybody is, nobody forces this is the way it should be done. I mean in the end our BA and the customer representative have to finalize it, but if it is in the development team nobody, I mean everybody is taken equally to come up with ideas and come up with the creativity (C01010).

However, it was noted that although someone comes up with an idea it must fit into the current scope of the project, so there is no limitation on idea generation or creativity, but “at the same time, it may not become a part of the project, or maybe not be put into the scope depending on the requirements of the . . . the requirements at hand, so” (C01010).

Although autonomy is crucial, it must be approached with a willingness to seek input and advice from other team members. So, although a member may be working on something by themselves it is still a good idea to enlist feedback from their counterpart at the other location or another member of the team (C01002). Overall, the members of this team appeared to open to input from others and giving and receiving encouragement to come up with new ideas and better solutions to problems:

> It’s very . . . our team is very open to encouraging people to come up with suggestions, better ways to do something, or if something is not effective enough then we come up with ways to make it more effective, so there is a lot of brainstorming that is going on amongst everybody in the group. So there is no, you know, there’s no, you know, . . . our project manager just makes sure that we are, you know, doing everything in a more time efficient manner, but everybody is always . . . every time you have a phone call or Net Meeting everybody is encouraged to give their feedback. Even when we have our demo, we do it like a retrospective after the demo and we all chime in on what good things were accomplished, what things could we do better, so we are always finding ways or ideas in which to make the project more efficient, you know, so . . . everybody has an input that is important (C01006).

Due to the critical nature of the projects that this team develops it is so crucial that autonomy not be taken for granted. There must be regular consultation and consideration of how certain changes might affect the project. As one member indicated, “the people usually think twice before change something important, they ask other people what the consequences might be and get much more involved into the project because of this” (C01007). Although individual decisions may be made on inconsequential matters of the project, It is important that no one person makes a major decision on their own. So the decisions are made at different levels as suggested by another member on the team:

> Well, it's at different levels if you look at . . . when you . . . when a person is given free reign they would implement something which they feel is good and something new or something that they want to explore and sometimes what they want to explore may not fit in with the larger thing, you know, in terms of maintenance and something, so sometimes they may have to as a group kind of decide which is good (C01008).

Finally, to some degree autonomy may be limited by organizational decisions or schedules that the team does not have direct control over and thus must make adjustments within the broader organizational structure:

> I really would like to come with some idea and suggest some new solutions, unfortunately, I have to say it, you know, when you are working in some corporation, in some big organization, there are different, there are a lot of number of differences in schedule that comes just from the outside, like the outside because it comes from the company but not from our team. We just need to find some way how to work with some company's really . . . really schedule and still follow some agile practices, you know (C01009).

Feedback. In this particular team the influence of feedback on the configuration of the team is very important:

> It's the next one, I believe it is a crucial thing. This is the one part that creates . . . this is one of things that creates the team to be committed to the project because

> without being in track with what is going on in project we would not have team . . . we would have surprised team members. I believe this is crucial (C01003).

It is important for the team to receive feedback from multiple sources. On the personnel side, the team receives feedback from upper management, project management, business or market advocates, individual team members, and customers. On the project side, the team receives feedback through various agile development practices such as the daily stand-up, iteration demos, and iteration retrospectives, and various tests. As one member commented, "every two-week iteration demo gives us a lot of feedback. Our acceptance test, our . . . even our unit test gives us feedback. From managers you know we just receive some praises" (C01007). Another member also alluded to these multiple sources of feedback:

> Talking about day-to-day . . . we have demos, so . . . two-week demos, two-week iterations and after each iteration we have a retrospective. So, we know what we have done and what not, we have also some tracker log so we know if there are some bugs within what we delivered, so this is on a continuous basis and from time to time we get some feedback that we did a good job from the product manager or project manager (C01005).

Although on a daily basis the team receives regular feedback from the project itself and the project management team, this member talked about the importance of receiving input from upper management as well:

> I think it is equal importance because you want to hear from the higher ups that you are doing a good job and you also want to, you know, basically get . . . especially, you know, good kudos, I mean our team we've gotten two . . . I guess

> two awards, you know, for being the bigger, better, cheaper, faster team and for being innovative, so when you get that positive feedback from the VP and Senior VP, you know, that's like a pat on the back, it's like everything we have done really pays off and they appreciate what we have done (C01005).

Adding to this statement the member continued, "but, I think, you know, even if it comes from your manager you need sort of "thank you" or "you guys did a good job", you know, that makes you feel good, you know" (C01006).

According to one member the feedback may be broken down into two levels. The first level involves the development phase when,

> We get the feedback from our sponsors when we do the demos, we get immediate feedback on that, how they are showing, whether they are showing interest in knowing or there coming up with questions, so we know the feedback and they do communicate how good the team presented, and sometimes they do that (C01008).

Included in this first level is feedback from the market or business advocate who deal directly with the beta customers:

> So a beta customer who is testing or using the application has an issue then they can call one of the advocate and the advocate goes with the project manager and the project manager tries to resolve the issue, if it's . . . and then she finds out whether it is a programming issue, if it's a training issue on the client's part. So the advocates are just like a middle person between the customer and the programmers and the QA team (C01006).

The feedback on the second level comes from the product manager:

> When we finish the project and . . . or when the product is demoed to a new customer we get the feedback and she distributes that to the whole team, the feedback she gets from the field and also like when a new project is done for a particular customer then the customer goes through the testing and then any feedback that comes from the customer is shared with the development team, so it is varying levels of feedback (C01008).

Core Norms of Conduct

In this team it was difficult to make a distinction between primary and secondary norms and so this section simply explores the source and influence of norms overall. In this team the development of core norms of conduct has been more evolutionary as members became familiar with each other and built positive interpersonal relationships:

> I haven’t seen any official norms or anything, it is kind of . . . it has evolved I would say as, again, I mean, as people got to know each other it evolved, rather than as defining a clearly cut documentation, okay this all we have do, this . . . haven’t seen that, I am not aware of that, we have not practiced that kind of thing (C01010).

According to Hackman (2002), core norms of conduct should specify “what should always be done in the team” and “what should never be done in the team” as an ideal. In this team, although policies are provided from an organizational and project management standpoint, one Polish member of the team felt that these policies were not the driving force for conduct, but rather developing norms more naturally:

> Well of course the policies . . . company policies exist, but in a team of reasonable guys like my team is, the Polish is let's say are created by life . . . we just develop them in natural way. I don't think we need some policies from company, we could live without it (C01007).

Similarly a second member suggested that due to the size of the team which averages around twenty members, the team relies more on regulating themselves much like a class or family where everyone knows everyone else and ". . . in my opinion we don't need the company policies to behave correctly and don't hurt anyone" (C01007). In addition, there is a central person located in both sites to oversee that members are following the necessary guidelines and that communication channels are kept open. These positions are held by the project manager in the U.S. and "we have a senior person there, the [Poland] folks report to her, so we copy her so she is in the loop if we need certain things to be followed up" (C01008).

In general, the attitude of the members to the influence of the norms of conduct on daily activities was very positive as communicated by this member:

> Somehow we have been lucky. Our managers are not, they do step in . . . we did have some sort of, kind of guidelines, but pretty much that was it. As to, I mean, those guidelines, it was never like "oh, you have to do it, are you doing it or not?". (C01008).

Team members understood that when problems arose outside of the team the must seek the proper organizational resources, but disagreements between members could be addressed within the team itself via a conference call and discussion as indicated by these members, “if we disagree within the team we have to resolve it our own” (C01005). Similarly:

> I think we have just found a way to address them. I know as far as if there is something that has been discussed and a person doesn’t understand then one of the team members interjects and tries to reword it to where it can be understood, you know what I am saying. They reword the question or they may elaborate on what is being asked so that both parties can understand, you know, what the issue is. I don’t think that it is a policy per say, I think that it is just the individual who is willing to help the other team members understand what’s being ask or what’s being answered (C01006).

There appeared to be an overall openness to member input and the ability to stop discussions when necessary and come back to ideas later on in the development process. As mentioned by the member below, there are some guidelines for the daily standup meetings. A major consideration is the duration of the meeting. According to the project manager the daily stand-up is scheduled for 15 minutes and should not exceed this if at all possible:

> When it comes to standup meetings and those kinds of things, there are a set of guidelines that we follow for that part. But, on a normal, like a brainstorm session or iteration planning, anything . . . if something from time limit or, I mean, if anybody feels it is taking extra time than we want to discuss, then everybody had a say in okay, let’s stop this here and then take it up later. So it’s not like one person who does . . . everybody has a voice to say. Nobody would limit anybody to that (C01010).

Team Composition

Size. One member got straight to the point in regard to the size of the team and suggested a “dream team” configuration for globally distributed agile teams, “my dream team is up to eight developers, something like three, four testers and two, three BAs and this team can do everything” (C01007). Overall, the members of this team leaned toward a moderately sized team, although sometime it is difficult to define the terms, small, medium, and large as the following member pointed out:

> My hunch is that the middle size, I don’t know what you call, what you mean by large, medium, and large, our organization, we’ve got about, including the testers here, we’ve got about ten here and average of ten or twelve or so in Poland, I would consider that kind of a medium size team, I guess. It seems to be the right size for what we have been asked to do (C01002).

During this research project this team consisted of twenty-one members. According to another member of the team, “I think if you have a large team you run into more problems and issues. But in my experience smaller teams work best” (C01006).

The complexity and size of the project also plays a part in determining the appropriate team size. There needs to be a balance between having too few and too many members on the team so that the team is not overwhelmed by the project, but on the other hand, the project progress is slowed down because the tasks are divided up among so many members that others are having to wait. The team size helps to measure what the team thinks it can realistically delivery in a certain time period.

> I, you know, it would go back to the size of the project and how many people is too many people, you know, at some point you know working on a project, you

know, so there is that delicate balance about moving people across projects and you know, at some point, you know, the size of the team, it is a very delicate team, but we have had the opportunity of having a lot on our plate where we can actually pick, okay what is the right size . . . what do we want to take on our plate and how do we distribute that team in a way where we can deliver when we need to deliver and not kill the people . . . (C01004).

So, a limit to the size of the team and does appear to be necessary “and when you exceed that limit it might have a negative impact on the project . . .” (C01005). However, related back to size and complexity of the project, if there is enough work and the team is moderately stable in its lifecycle, “I think the limit may be little bit higher (C01005). Subsequently, simply adding more members to the team does not necessarily make it more agile and able to accomplish more work:

> If you add more developers it doesn’t mean you can do things faster because of many things, for example, when you start assigning more and more stories, the probability that stories overlap somehow are dependent, the probability increases and in a way someone must wait until someone else is done with something (C01007).

Increasing the number of members appeared to have a negative impact on agile practices such as iteration planning and the daily stand-up meeting:

> In agile approach . . . it is something I thought about sometime ago and what I found is that for agile approach to have too big of a team is not good and many practices will not work correctly. I can use some specific number. There was a moment in my team when I had in [Poland] more than 20 people on the team. It

was difficult, it was difficult to do a good demo . . . not demo, it was difficult to do a good iteration planning meeting. It was difficult to do a good stand up, and it cause a situation that when I have to many people in the meeting not too many of them were involved and it was very difficult to keep the big team to be really involved, smaller teams are better motivated, are better involved in what is going on, they are feeling more responsible for the project (C01003).

This sentiment was echoed by another member of the team:

> I think that too big teams are not good for agile. Because of . . . again, a few things. Let me start, for example, stand-ups which are very important when you do communication between the whole team. We grew up to, I don’t know, it was about twenty people here and in [United States], yeah, it was twenty people, meetings became ineffective, they took too much time, always someone started talking about something irrelevant, just the meetings . . . the stand-ups loses their focus, then the team lost a few people and additionally did . . . we started dividing stand-ups into sub-sections and things went again well. The stand-ups started working as expected (C01007).

Consequently, one team member made the following suggestion, “so, I believe for agile it is not good to have bigger than twenty, let’s say twenty people in the team and for bigger project I would suggest just to subdivide in project and sub teams” (C01003). Because of the importance of effective interaction between members, a consequence of creating a team that is too large is that “the collaboration and coordination becomes difficult, it makes it difficult, and also like you want to communicate something, it’s making everybody agree upon something makes it a little difficult also if the team was bigger" (C01010). A smaller team may also help alleviate cultural differences in that all the members know each other ". . . from where they come from and pretty quickly members learn how to behave and this is not the case. If the team has I don't know fifty people it would be a case" (C01007).

However, a couple of team members did not rule about the possibility of using an agile method with a larger team as long as there is a point person at each of the distributed sites to resolve any communication issues that arise:

> I think it can be very large, I don't see why it cannot, as long as you have somebody senior in each area where . . . to resolve anything or whatever. I mean sometime when you are crashed down they cannot wait because of time zones to get questions answered and it becomes a little . . . communication also, I mean, it's nothing like face-to-face communication, you need to have some sort of a lead in each area, yeah. That helps (C01008).

A second suggestion for larger teams was to make appropriate modifications to the agile practices, ". . . but you need to be aware of that, that when you're team is growing up you need to change some things . . . because first you need to change something and to not follow any practice without thinking" (C01009). So, even though it is one large project, it can be broken down into multiple parts with multiple sub-teams with multiple iteration planning sessions and stand-up meetings:

> What to change, basically, we tried to split inside the team, split some enhancement we have or project inside to one project into some separate tasks, assign to this project some developers, testers, QAs, and analyst is nice and start different, for example, instead of one big planning sessions, start three

> separate planning sessions . . . and also doing stand-up, we have it still now, because we found out that it’s better divide stand-up into some tasks at the beginning . . . (C01009).

In sum, this member concluded that even larger globally distributed teams could be facilitated, “I believe that some changes to the agile when you have a really big team need to be done and it’s not matter if this team in two different places or not” (C01009).

Mix. This team welcomed diversity among its members, “okay, so I would say that I would prefer to have rather diversity team for sure (C01003)”. Creativity, generation of ideas, and shared knowledge were identified as some of the benefits of having a balanced mixture of members. And indeed this team does have a diverse make-up in regard to nationality, gender, age, and role all of which contribute to its uniqueness as indicated by the following statement:

> No, I think the differences are what make a team interesting, you know. I can’t see a lot of people who are the same on this team. We’ve got people from several different countries, we’ve got from 22 to 60. It’s, you know, a wide range of diversity (C01004).

A second member echoed these sentiments in the following way:

> I think it . . . I think it is important because if we are all developers the same age doing the same job I think . . . I mean everyone is different in the kind of way make, I don’t know, makes you feel safe or make you look from different perspectives, you know, my feeling is that it is . . . now, if there are too many

developers and little testers, too many men and too little women, you may have a bad impact on the team (C01005).

This combination of diverse members helps to mesh new and existing members who are all at varying levels of experience with the product, with the technology, and with the overall development process:

> if team is diversified taking into account, you know, important factors like, you know, experience with the product, inexperience with the product, experience with technology, inexperience with technology, men, women, older, younger, I think if they are diversified within this . . . within this factors it's okay (C010050.

Task and Knowledge-Related Skills. Overall, the team stressed the importance of task and knowledge-related skills. In fact, one member indicated that an agile approach contributed to getting members up to speed in a timely manner:

> For sure they are important, however, I would say that something that we discovered some time ago in this agile approach that agile brought very . . . new team members to be fully efficient in the team in very short period. Even, of course, it's important to have well skilled people and with great knowledge and task, however this approach we are following supports the situation when we have new team on the team who are not so experienced, but they could be efficient on the team also (C01003).

Teamwork and collaboration are hallmarks of agile methodologies. As such, there was broad agreement amongst the members that working to build on the strengths of each individual was important as well as compensating for members who might be weaker in a specific area.

On the whole the team advocated helping out when other members are not as strong in particular areas. When new members are hired for the team or interns are converted full-time members that are not expected to know everything, there is a learning curve involved when joining the team. There should be a balance between time spent helping others and time spent working on assigned work:

> I think if you have this kind of dynamic that everyone is still getting some skills from other people, I mean, provided it's not too much time, they don't spend, you know, 60% of time learning new things, and 20% of time doing their jobs, but, if you let's say you spent 20% of time learning new things from other team members I think, it's a good situation because you learn from each other, you are forced to communicate with other team members, you know, I think it is a good situation (C01005).

This is a high-functioning team with high expectations for its team members. As the project lead commented:

> I think it is important to challenge everybody and expect, you know, the most out of everybody, knowing there is going to be some learning that needs to be done, but you know, so it makes their job interesting. But, no I try not to . . . well, I try not to let people rest on their laurels for one thing and give them a challenge and kind of spread that around, kind of spread the challenge around (C01004).

The mix or diversity of the members is also related to knowledge and task-related skills in that knowledge can be shared among the members and those with natural talent can help to teach other members as well as help assist them when they are having difficulties:

> Yes, yes, definitely, that’s why mixture is good. You cannot be good in everything. There are people with natural skills for some areas and they can somehow cover this area where the other team members cannot be as productive or as good as this one. Also, this, he or she, can even learn other people if it’s possible of course. I am telling about some natural talent like painting or like that. If you have, for example, a good knowledge about test driven development you can learn other people (C01007)

With a good mixture of members it is not necessary for everyone to be an expert in all areas as long as the team member is motivated:

> In my opinion I don’t think everybody has to be strong in technically and functionally in both areas to be a successful. I think it could . . . you can be success even . . . as long as you are motivated to do, you don’t have to be too strong to finish the something on the project, I would say (C01010).

Interpersonal Skills. The interviews from this team indicated that interpersonal skills are very important in a globally distributed agile team. Agile methods are predicated on the concepts of interactions, people, and collaboration. The agile practice of daily stand-ups is one specific way that members interact with each other on a daily basis. Because this is done via ICT rather than face-to-face it is important that each individual have strong communication skills to get their point across to others, including the customer, “it is important, especially in agile I think. It is important where we have daily stand-ups, where we have discussions in the whiteboard where we design, you know, present our jobs to customers (C01007). This is further evidenced by the following comment:

> Oh, it is huge. Yeah. And I think, again, because it's . . . the agile process is such a collaborative effort that . . . that builds those relationships, those interpersonal skills because you are not talking to somebody every two weeks, you are talking to somebody every day. You are understanding what kind of road blocks are heading . . . people are trying to help each other, you know. They really do try . . . they really do try to help each other when they are seeing them . . . you know anybody struggle or you know if there is a problem then everybody is wanting to jump in and help, so I think that is very important (C01004).

Communication drives globally distributed agile teams and without efficient and effective communication, the team will have difficulty completing its projects as this team member indicated, "oh, it is very important, very important. Communication is what makes this work and it's . . . otherwise it becomes a drag on the whole group, for the people around them. It is important "(C01009). In addition having well developed interpersonal relationships between members takes on even more importance in globally distributed environments due to the cultural and language differences that may cause misunderstandings. As this team member stated, "it is very important to know each other interpersonally because that will help understanding or interpreting others when they are communicating or understanding the way the person is communicating, I think it is important (C01010)". Thus, having a more personal knowledge of the people you work fosters stronger communication, greater understanding, and clearer interpretation. Table 61 provides a summary of the findings for structure for team C01:

Table 61

C01 Team Structure Summary of Findings

Team Structure	Findings
Task Design	
Meaningfulness	Part of something larger, understanding the big picture
	Multiple concepts: importance, usefulness, quality, accountability, responsibility, understanding, ownership, trust, motivation, growth, and challenge
	Importance defined by usefulness to the intended user rather than by high priority or marketing priority
	Ownership takes precedence over importance, i.e., analogy of the parent-child relationship, the project ". . . it's like their baby"
	Give somebody something challenging to do which shows trust which instills accountability
	Meaningfulness, motivation, and quality are connected to a clear understanding of the overall goal

(table continues)

Table 61 *(continued)*.

Team Structure	Findings
	Establishment of a new organizational path, e.g., implementing an agile methodology to migrate from an existing legacy system
	Distribution of work: As the intersection of agility and structure agile practices such as short iterations and small releases can contributed to the meaningfulness of the project by alleviating feeling of being overwhelmed by user requests
Autonomy	Crucial and important
	Afford to small groups within the team to determine best approach
	Related to prioritization and scope: if change does not significant impact scope or prioritization afford a high level of autonomy
	Related to level of experience and expertise: new members need more oversight than senior members

(table continues)

Table 61 *(continued)*.

Team Structure	Findings
	Members need the freedom to implement appropriate practices for their specific tasks especially in distributed environment structure, decisions, or schedules
	Related to ownership and accountability: potential that one part of the team will lose a sense of ownership, especially if certain types of members are located at only one site or the majority of certain types of decisions are made only at one site
	Suggested remedies for address loss of ownership is strong communication and splitting ownership across sites
	Related to accountability and shared responsibility: more autonomy equals feeling more responsible
	Related to technical and business decisions: flexibility in terms of technical aspects left to a large degree in development portion of the team, i.e., as long as code works necessary changes can be made and different tools can be used

(table continues)

Table 61 *(continued)*.

Team Structure	Findings
	Related to scope: idea generation and creativity should not be limited, but not all ideas will fit into the scope or requirements of the project
	Related to other members: autonomy should be coupled with the willingness to seek input from other members; members should give and receive encouragement to come with new ideas and better solutions
	Do not take it for granted - there must be regular consultation and consideration of how changes may potentially impact the project as a whole
	Relationship to organization: may be limited by organizational structure, decisions, or schedules
Feedback	Feedback is crucial because it creates commitment
	Multiple sources on the management and personnel side: upper management, project management, business or market advocates, individual members, and customers

(table continues)

Table 61 *(continued)*.

Team Structure	Findings
	Relationship to agile practices: daily stand-up, iteration demos, iteration retrospectives, and testing are all sources of feedback from the project itself
	Members want to hear from all level of management, including upper management such as the VP or Senior VP, that they are doing a good job
	Two levels: Sponsors (including market or business advocate) at the time of the demos which provides immediate feedback Product manager distributes feedback from the customer and from the field
Core Norms	Evolutionary approach to development of norms - no “official” norms
	Organizational policies and procedures not the driving force for behavior, rather norms were developed naturally through interaction between members over time
	Relationship to team size: size of team facilitated regulating themselves as a class or a family where everyone knows each other

(table continues)

Table 61 *(continued)*.

Team Structure	Findings
	A benefit was having a person located at each site to help handle any conflicts
	Management was flexible in the enforcement of norms
	Resolution of disagreements between members was done within the team itself
	Overall openness to member input
	Relationship to agile practices: guidelines for daily stand-ups, e.g., scheduled for 15 minutes and focused on technical aspects of the project; no limitations on brainstorming sessions or iteration planning
Composition	
Size	"Dream Team": up to eight developers, something like three, four testers, and two, three BAs
	Dependent upon complexity and size of project
	Helps to measure what can realistically be delivered in a certain time

(table continues)

Table 61 *(continued)*.

Team Structure	Findings
	A limit to size is necessary as there is a point at which the law of diminishing returns sets in
	Agile, no more than 20 members, then start dividing into sub-teams
	Small team:
	Overall members leaned toward smaller team size
	May lead to feelings of being overwhelmed
	Smaller team may help to alleviate cultural differences in that all members know each other better and are able to build stronger relationships
	Large team:
	Run into more problems and issues
	May slow down the progress if tasks are divvied up too much and members are having to wait on other members to complete their part
	Simply adding more members does not necessarily increase the productivity of the team or make it more agile

(table continues)

Table 61 *(continued)*.

Team Structure	Findings
	Increasing number of members negatively impacts certain agile practices such as iteration planning and daily stand-up meetings, e.g., stand-ups start losing their focus
	Makes communication, collaboration, and coordination More difficult
	Need to have a point person at each location for communication and coordination purposes
	Necessitates the need to modify certain agile practices
Mix	Diversity welcomed
	Benefits included creativity, generation of ideas, and shared knowledge
	Helps mesh new and existing members who are at varying levels of experience with the product, technology, and with the overall development process

Team Virtualness

Extent (Degree) of Virtualness

For this study team virtualness was defined as the extent of virtualness of the team based upon the characteristics of temporal distribution, boundary spanning, lifecycle, and member roles. The following section provides detailed analysis of the team virtualness dimension and its related sub-dimensions beginning with the extent (degree) of virtualness. Based upon the characteristics of each sub-dimension this team represented a medium level of virtualness as summarized in Table 62:

Table 62

C01 Extent (Degree) of Team Virtualness

Sub-Dimension	Comment
Temporal Distribution	Distributed across the United States and Poland
	Time zone difference 7 hours
	Overlap 1 hour
	Multiple ICT were employed
Boundary Spanning	Crosses functional, organizational, and cultural boundaries
Lifecycle	This is an ongoing project team which adheres to a longer lifecycle
Member Roles	Roles are well-defined and for the most part members do not play multiple roles
	Members are allocated 100% percent to the team

An examination of the sub-dimensions of virtualness indicated that C01 did have at least one hour of overlap for potentially conducting daily-standup meetings, iteration planning, iteration demos, and iteration retrospective. Although the overlap was minimal there was at least some time for synchronous communication. To its benefit C01 also utilized multiple ICT to improve its capability for communication.

In terms of boundary panning the team crossed functional, organizational, and cultural boundaries. Functionally speaking the team worked closely with the marketing area as well as other functional units. Considering its size crossing organizational boundaries was cited as a given. Finally with members geographically distributed in two countries several cultural boundaries were identified. In regard to lifecycle the members preferred a longer duration. The members participating in the study were stable with all having worked on the team for at least one year, with many others working for 3 or more years. Some turnover was cited, but by and large the team has maintained the same members. Finally, in relation to member roles, the organizational employees were open to playing multiple roles on the team when necessary and did not feel that it negatively influenced the team configuration. However, roles were typically well-defined and members were allocated 100% percent to the team and did not play roles in any other teams.

Temporal Distribution

Benefits. One of the most cited benefits of temporal distribution is the potential for a 24x7 shop due to the time zone differences. This was echoed by a member of the team who commented that work can be handed over during the daily meeting as the

U.S. team is finishing the day and the Poland team is starting the day. This also provides continuous, round-the-clock coverage of the project. "If the build gets broken there is always somebody that's available to look at it and get us up and running again. So that's a big benefit" (C01004). In order to take better advantage of the temporal distribution in relation to builds the team agreed upon procedures to better manage the submission of the builds “like no check-in at the end of the day, just check-in before, so those kinds of things we followed, so, I mean we could overcome some of the small, small issues” (C01010).

Temporal distribution and time zone differences in essence add time to the work day. As one member suggested:

> I mean, the benefit to me is over a two-week period you’ve got, you’re effectively getting, let’s see . . . two weeks . . . here you’ve got ten work days, because you are not all working at the same time, I think you are getting a little bit more out of each work day because of what the team in Poland is able to get to work on during their eight hours, we’ve got this one hour opportunity to hand-off, for the team here to pick-up, basically you are getting two days, it is probably not two full days, but you are getting the better part of two days in a one day time period, so I think that is a significant benefit as long as that overlap happens and as long as the communication and follow-up and all the other back and forth that is needed sometimes, sometimes the developer level, sometimes it is project management level, as long as that is happening, I think the benefit outweighs the challenge (C01002).

This ability to “hand off” problem issues to the team which is beginning its work day allows additional time for that problem to be solved by the time the other team begins its work, thus:

> . . . the solving this issue could be continuing on the US business hours and when we are coming back to the office, the situations could be solved and the same in other side, when they find in American business hours they could be solved during Polish time (C01003).

Consequently, the daily “hand off” provides the benefit “that if something bad happens and [U.S.] sleeps then we can take an issue in Poland and deal with it. If we are in our beds then [U.S.] works” (C01007). So, this element of temporal distribution was considered the most important benefit by one member of the team because “so it is good to have teams in different places so we will be able to answer on that issue faster, this one benefits. And I think is the most important” (C01009).

A second benefit cited was the cost savings associated with offshore locations, “also, I don’t know, probably from the [U.S.] perspective the [Poland] team is cheaper so this is the benefit for company” (C01007). In regard to the use of an agile methodology specifically within a globally distributed environment one member stated, “. . . there are challenges for sure and there are benefits both ways. But I think if we were to try to do this in waterfall it would be a lot more difficult, a lot more difficult” (C01004).

Challenges. One of the most commonly cited challenge in globally distributed teams is communication. In order to alleviate some of the challenges of global distribution the team stressed the importance of the regular meetings, “yeah, I mean there is a challenge with communication which is helped by the standup meeting" (C01004). To improve the effectiveness of the daily stand-ups rules have been established for keeping the meeting to a maximum of fifteen minutes and enforcing a rule about staying on topic.

More specifically the absence of face-to-face interaction was identified as a major issue in regard to communication. As one member stated, "the challenge is when there is absolute no communication, visual communication, visual interaction between two teams then it becomes a little difficult" (C01008). Due to the complexity which can be involved in software development finding and implementing the necessary tools for facilitating effective communication may also pose a challenge when face-to-face interaction is not possible, "when you don't have perfect tools that may make face-to-face communication you need some, you know, takes more time to explain the ideas and sometimes you just don't get information, you just . . . you know . . ." (C01005). In light of this, if possible If possible having select members from each location visit the other was suggested as a way to improve communication, build personal relationships, and foster trust:

> So when you meet at least once or twice and especially not just the very beginning, after you have worked a little while and you have already formed some ideas about your teammates and then you meet with them, after that a lot of things go smoothly, very smoothly, you've earned trust and they've earned your trust and you've also put a face to, you know, the changes being made by that person, so that goes a long way. So, I mean, when you speak with

> distributed, that works very well when the team has had a chance to meet with each other at least once or twice . . . (C01008).

Along the same lines, another member of the team said the following about the benefit of on-site visits:

> One person came here for a couple of weeks and the other ones been here for two months to build that relationship and kind of understand how things are done here and we've had people go over there for a month or two at a time which has really helped as well, to understand the culture in the different location (C01004).

Global environments also lack the informal communication that often takes place in colocated teams, "you pass some information just drinking coffee or going for lunch, you do it, you know, without any efforts, just while doing other things and if you are in two location you have less possibilities to pass this information" (C01005). Unfortunately, at times global teams fall prey to the "out of sight, of mind" trap as suggested by the statement made by the following member:

> so, that is . . . and it's not anything intentional it's just easy to get up and somebody is in your space and say "hey, how about this or how about" and not remember to include, you know, to remember, you know, the team on the other side ocean. So that, that is a difficult . . . that's difficult (C01004).

The distribution of the project management team was also identified as a potential challenge related to communication. For this team in particular at the onset there were two project managers, one located at each of the sites:

> I think it was a little bit more difficult when doing it that way, but now that the team has merged into one team, we've actually had an effort under way this year

> in their objectives and goals for this year is improving communication and we've met and talked about what that looks like and what that means. (C01004)

This could be related to having a clear understanding of who is guiding the direction of the team and who is responsible for making final decisions. However, another member emphasized the feeling that there should be a distinction between the project management side and the development side, so that each would be able to focus specifically on their tasks and thus "this is a kind of split of responsibilities so this could be a benefit" (C01005).

In addition to communication issues related to geographically distributed locations, time zone differences are a second commonly cited challenge for teams. A conscious effort must be made for scheduling meetings which take into account the working hours of both locations because working "in different time zone you need some efforts to plan meetings so they . . . so we are able to have a meeting with other folks" (C01005). Because there is at least some overlap of working hours, meetings are scheduled during this time and thus "nobody is angry that has to go very early to the office or stay late, yeah" (C01007). For dealing with the time zone difference it was also suggested that members consider working from home:

> Actually, I think if you are working home it's easier to work with guys in different time zone. It is more naturally, we make a break, you go to sleep, you get up, you work again and if you are . . . you have this eight hour, let's say you start at nine and end at five it's little bit harder to find a common time window and if you work from home and the other parts work I think it is easier to be flexible (C01005).

Time zone differences may also hinder the development of a sense of “teamness” among the members as the following statement suggested:

> if you have one team in one time zone you are just working just one team and if you have two teams in two locations you are constantly trying to make one planning session to make everything together, but if you don’t do any efforts to keep these two parts together it just . . . you result in having two teams and if you have them in one location you don’t . . . they are just one team (C01005).

It is important for the team so see themselves as one team rather “than just colleagues working across the ocean” (C01002). Perhaps one strategy for fostering this sense of “teamness” is through the idea of shared ownership, “there’s not one component that is solely owned by the team here or the team there. There’s ownership from, you know, from both locations across all of the product . . .” (C01004). Shared ownership may instill the attitude that the team is truly working together in a collaborate manner. See Table 63 for summary of temporal distribution benefits and challenges:

Table 63

C01 Benefits and Challenges of Temporal Distribution

Benefits	Challenges
Time zone difference allows for 24x7 shop	Communication issues necessitates importance of regular meetings
Handing off work to the other team at the end of the work day	Absence of face-to-face interaction
Continuous, round-the-clock coverage of the project, adds time to the work day	Lack of informal communication over coffee or lunch
Cost savings	"Out of sight, out of mind" trap
	Distribution of project management
	Understanding who is guiding the team and responsible for making final decisions
	Making a distinction between project management side and development side
	Time zone differences cause scheduling problems for meeting

Use of Information and Communication Technologies. Due to the importance of communication within a globally distributed agile team this team utilizes multiple synchronous and asynchronous information and communication technologies (ICT) to facilitate interaction and dialog for such regular scheduled meetings such as the daily stand-up, demo presentations, and iteration planning (C01003). These include commonly used ICT as teleconferencing, instant messaging, desktop sharing, project management tracking, and email. The team has also experimented with videoconferencing and the use of tablets, similar to whiteboards, for drawing diagrams which can be viewed remotely.

The team has setup bridge lines for utilizing regular teleconferences and member-to-member calls. Daily phone calls are a common occurrence and serve as a primary means of communication for the daily stand-up. As one member commented, "yes, oh yes, teleconference, we use that . . . hour to hour basis I would say. Phone conversation is very much in the picture" (C01010). Similarly a second member reiterated the importance of voice communication, "now, yeah. And the phone, I mean if we have to talk, we talk, we pick up the phone and call each other" (C01008). The primary limitation of phone communication is the time zone difference between the two sites which depending on daylight savings time provides only one to two hours of overlap (C01008).

Instant messaging is also a regular ICT in use by the team members for synchronous communication both for inter-office and between distributed sites. Because of its capability to support desktop sharing it is also used for iteration demos, iteration planning, reviewing code, and exchanging ideas (C01010). Similar to telephone communication instant messaging is limited by the overlap in working hours. As one member commented, “instant message is used fairly regularly. Again, once they leave the office and go to bed and it’s noon here for me, my instant message is fairly ineffective, I may as well as just email” (C01002).

In addition to the desktop sharing functionality provided by the instant messaging tool the team also uses various versions of collaboration software on a regular basis. In particular Microsoft® NetMeeting is used specifically for demonstrating iteration demos, discussing user stories, and conducting iteration planning (C01006; C01007; C01010). In regard to the use of NetMeeting one member indicated that it is “the most effective we are using because you are able to see each desktop and the setup is quite simple, so this works quite well” (C01005). Similarly, another member commented, “we can see, you know, when we do our demos you can actually see what they have completed with their code, so NetMeetings are helpful” (C01005).

An open-source application is used for storing backlogs and user stories and members can “bring it up on a NetMeeting and everybody’s able to follow that and talk to it” (C01004). In addition a spreadsheet is utilized by the project manager for providing a critique of how the team is progressing or to “give more input on stories, changes, or development, so just having that tool has been very helpful, with you know a team, you know, that is very diverse in that aspect” (C01006).

Videoconferencing is available but is not used on a regular basis. Several factors contributed to its occasional use. First is the expense incurred, “but, you know, we are not using this as needed because it also cost money” (C01005). Second is the difficulty in setup and use of the equipment. As one member commented, “the setup is too complicated, you know, so we are using this every two months . . . three months . . . so, you know, if things are not simple we are not using this" (C01005). Third is the difficulty in securing facilities at the Poland site, "we do have some and there's one room in the building that has that, it's hard to find the buildings in [Poland] that have it because they're used quite a bit" (C01004). Finally, a member on the team mentioned the presence of a possibility of the intimidation factor associated with the use of a new technology:

> I think the newness of videoconferencing was intimidating to people, we didn't find it particularly helpful, you know, it was kind of novel thing that you could see your colleagues on the other side of the world, but it did not benefit the conversation, we were all much more comfortable to sit with our headsets on and talk on the phone (C01002).

Overall, the use of videoconferences has been used primarily "when we are meeting with higher management, not during demos for example. But I don't think it would be very handy on demos since . . . the most important point is to show a product, not our faces" (C01007).

Drawing tablets similar to whiteboards have been used as well with mixed response. Although the tablets allow members to draw pictures and diagrams which can be seen by remotely by others and have the potential to be a helpful tool (C01004), several limitations were cited by members. First, the difficulty in ease of use and quality of the product:

> And we are also played a little with tablets, you know, and whiteboard from NetMeeting, but I don't know it's not so . . . maybe the cheap equipment but it

just doesn't . . . it is not so easy to use . . . it is not so easy to write, they promised you should be able to write on the paper, but, it doesn't work so well (C01005).

Second, the technical problems associated with the use of the tablets in a globally distributed environment, "it wasn't successful, probably because of the network bandwidth. The drawing on the U.S. side was too slow" (C01007). As indicated by another member, "we are also try, but it is not used practical, to use pads for drawing graphs or something like that, but for now we are just trying, it is not an effective tool" (C01003). As such, overall the team has not found the technology to be overly effective.

Boundary Spanning

Functional. In this team the IT departments works very closely with the marketing personnel. The fact that the marketing department has also adopted an agile process has been very beneficial and while this represents the crossing of a functional boundary, it has not been a hindrance to the team, but rather a plus.

Organizational. From an organizational perspective the only barrier that was mentioned was the difference in experience level between the U.S. and Poland teams. On average one Poland team member surmised that the U.S. team members had a greater degree of longevity with the organization:

> Well, so first . . . first boundaries they are . . . an average like twenty years with Organization C, and we are, I don't know three in average so they are older and they work longer for the company so this is one kind of barrier (C01005).

Cultural. According to the team members multiple cultural boundaries must be taken into consideration including: overtime, language and communication, sick leave, holiday schedules, and team size. Interestingly enough one member from the Poland team suggested that crossing cultural boundaries in the team has been less of a challenge perhaps due to the fact that “in American part of our team that our members have different nationality, so our team is really multinational. It’s maybe trained, but I think it is easier to have multi-national just to national for instance” (C01003).

In regard to overtime an American member of the team commented, “I have also noticed that they do not work a lot of overtime like we do in America” (C01006). A second American member recalled that when the two locations initially combined to form one team approximately two to three years ago:

> the average developer in Poland, when it was five o’clock in the afternoon in Poland, they were gone, because that was just the way they thought. You only worked eight hours a day and nobody expected you to work a minute more (C01002).

This concept of working hours represents a cultural and possible economic boundary between the two nations which initially was a challenge for the team. However, as the team has developed and has worked together now for a longer period of time, the same American member made the following observation:

> interestingly now several of them have started to work from home, several of them now don’t think twice about staying late, they finish their day, they will dial in for a conference call, or they’ll be on instant messaging, and I know it is ten o’clock there, but they’re . . . but we’re communicating, . . . it has been interesting

> for me to see that change happening, you know, within their little micro-society from where I am sitting. It used to be a problem, I mean, it used to be a problem and you could not get in touch with anybody in Poland after five o'clock their time and it is no longer a problem (C01002).

Working from home has also increased the flexibility of the members in terms of work hours and to some measure is "kind of breaking the boundary of the traditional eight hour day . . . it works as long as the two-week iteration is going well and everybody's happy with what's being delivered" (C01002). As a member from the U.S. team indicated that the issue of working overtime due to the time zone difference was primarily encountered during implementations, putting code into production, which needed to be done at certain time as not to impact the customer. Initially, the U.S. team was doing most of the implementation and testing. Recently, a change was made to gear up the Poland team to assist in supporting the "implementations and test efforts earlier in our mornings and their afternoons, so that's a benefit as well" (C01004). This strategy also reduces the amount of overtime necessary due to the time zone difference.

According to one member of the team, "language was not much of an issue even from the beginning because they only hired someone who could speak English. We haven't had much of a barrier there" (C01002). However, a member from Poland explained the following scenario as it related to differences in accents among the team members:

> I did not notice any particular issues with particular cultures, maybe, I don't know, during communication, in the beginning, but only the beginning only. Different

> people from different countries have really different accents and for someone who is not native and for example, in my case, English is not my mother language and when I talk with guy from India, not from my team, but from other team and I just hear his accent, I cannot understand him, but it really takes some time to seriously process it to the normal English and to my language and so on (C01009).

Communication as a whole did pose a bit of a challenge initially. As one member commented, “we did go through a communication challenge, I think we have solved that” (C01002). Another member referring to this same initial cultural boundary recounted some specific communication issues such as word meaning, word choice, interpretation, and individual personality and how they were solved by the team:

> I would say, yes, we overcame so, eventually, every . . . people in every country have different ways of communicating for sure. So, yeah, we, I mean sometime the choice of words could make a changing the meaning so that’s . . . we did notice initially and then sometimes you interpret differently and I think the other side is a little bit harsh on their explaining or aggressive (laughter) on explaining their idea. So, we had some difficulties in the beginning, but over the period we know what exactly like . . . what exactly the person is thinking, like when you go . . . when we visit there and they come here and so when go face to face, you understand based on the way he communicate in a general way rather than in a meting and you interact with the person, so you know, like what kinds of words . . . choice of word the person is using so depending . . . so then that will give you some kind of idea when you are in a meeting conversation when that person is

> speaking, so even though he may sound aggressive, but he may not be really that way. Yeah, we did have that initially, but as the team progressed and the people get to know each other, we kind of understood each other very well, after a certain time (C01010).

The diverse and the multinational characteristic of the team also helped to foster understanding in the event of language differences:

> I did find that the people in Poland because they would sometimes have to kind of pause, put it on mute and speak to each other as a group, a little group there in Polish to make sure they're on the same page us . . . yeah, so we understand that. That would be expected. And me I worked in England, I worked in India, and I worked here. I came here many, many years ago so I know what it is like on that side (C01008).

A member from Poland suggested that identifying areas of commonality could be a starting point for conversation and communication building, “if there is a NFL final or in the US, or you know, something that everyone is watching on the TV and you can just discuss it”, the member continued, “you know, on the meetings and there are not so many common events that happen both for American and Polish people, so if we had this kind of event it would be just easier to start conversation, you know” (C01005).

Another cultural boundary that is labor related is the handling of sick leave. Typically in the U.S. if people are out sick they will gauge whether or not to come in for work on a day-to-day basis. When they feel well enough they will then return. Evidently in Poland the labor laws specify that doctors may prescribe a pre-defined time that the person must be absent from work, ranging from several days to several weeks, depending on the type of illness with a three-day minimum. This was one of the cultural issues that the team has had to address. As one member of the team commented, however, “it is just interesting, I mean, there is nothing necessarily bad about it, it is just different, it is just different from how we do it” (C01004). Echoing the influence of culture on a globally distributed agile team, another member suggested, “but again that is a cultural thing, their history is just different and their economic system was different, their expectation was different” (C01002). Of course, anytime any member is absent from work there is a chance that the agility of the team will decrease simply because of fewer resources to work on the project.

In regard to the affect of holiday schedules on the team, a Polish member of the team, stated, “it is not an issue” (C01003). From the perspective of an American member their comment was simply in reference to the difference in the number of holidays in Poland, “yeah, I know Polish . . . the Polish folks get a lot more holidays off than we do, which is not a bad thing, I would like to have that many holidays off too!” (C01006). Another American member referred to several religious holidays spread over four, non-consecutive days which the team had to initially work around. As the member recounted:

> The first time we all had to deal with it, it was a little bit tricky. Because they kept saying “we are out on holiday tomorrow”. And, you know, at first we didn’t know what they were talking about but know we kind of know okay we got this little . . . slow period in April or March whenever it is we have to watch out for (C01002).

The key to addressing holiday schedules is upfront knowledge and appropriate planning on the part of members in both locations. Overall, this type of cultural boundary spanning did not pose a significant challenge to having members located in the U.S. and Poland.

Finally, a member commented that team size may have an influence on the impact of cultural differences in the sense that in smaller teams members are more familiar with each, because they know ". . . from where they come from and pretty quickly members learn how to behave and this is not the case. If the team has I don't know fifty people it would be a case" (C01007).

Lifecycle

There was mixed response among the members in regard to the length of the team lifecycle. The members were honest in their assessment one stating, "I don't know, I don't know how . . . whether the long period or not for the agile team" (C01003) and a second, "so what I prefer, short or longer term, hard to say" (C01009). One issue in particular that has been a challenge for the team is "trying to find a couple of skill sets over there in [Poland] which is the business analyst and quality assurance, it is not a . . . there is not a large pool there for those types of resources" (C01004). Consequently, for those positions in relation to lifecycle, the team would like to retain those individuals as long as possible. A second issue related to geographic location is the growing marketplace in Poland, as one member commented in relation to the Poland team, "I don't see a lot of movement, we've had a couple of individuals who have left the company, the market is pretty challenging over there, it is very competitive as far as wages, so that was not surprising" (C01004).

The emphasis for shorter a lifecycle was primarily based upon the need for bringing in new members to keep the team from growing stale, comfortable, or complacent:

> I would say that I believe for team it is good to have changes, to allow to go to different teams, although new people join the team, it is important, it refresh, it diversifies the team, it allow bring people, bring new ideas, generally it something very, very good to have . . . to have . . . to take new people into the team regularly, to take new people we also allow some people leave (C01003).

As another member commented, “gaining members is certainly good, you get new ideas, fresh ideas and you have enthusiasm that they bring into the table which is very, very good” (C01008).

Additionally it was suggested that by keeping the team lifecycle short and rotating new and existing members that the expertise of more knowledgeable members could be shared across teams:

> I think it is good to get new blood and to have the people who really know it move to other teams to, you know, help . . . to use that knowledge in the other teams, but there are a lot of new people there that don’t have that knowledge and it helps a bunch, you know, and see it in work . . . to see those practices in work and how they benefit the development process. (C01004)

However, the rotation of should be done in a balanced manner because if too many new members are added too quickly it can be detrimental to the team as is the case when too many experienced members are lost to other teams “because then is a drain on the people to answer questions or to help somebody else out” (C01008). There was even a recommendation for the combination or balance of new and existing members who should be on the team at given time, “I would say that to combine those two things, to have two people being longer perspective and allow new people to bring new ideas, new perspective is really good” (C01003).

Those advocating a longer lifecycle do not necessarily argue against the need to bring in new members for many of the reasons stated above, however several reasons were provided to support members staying on the team for longer time periods of time. As this member commented, “so I think a longer life cycle is better because everyone, you know, is pretty much knowledgeable and they’re focused on what needs to be completed” (C01005). Because of the complexity and size of the projects developed by the team another member indicated,

> Yes, I think if you are staying longer on the team it is more beneficial, because these are . . . I mean, in our company there is a steep learning curve so you need to learn a lot of things so if you are staying, you know, a few years on the project, one, two, maybe three years on the project compared to if you stay half the year or one year on the project it’s . . . it’s more beneficial to stay three compared to one year in the project (C01005).

Communication, knowledge, and familiarity with each other and the project were cited as important reasons for a longer lifecycle. All of these aspects are primarily developed over longer periods of time:

> Yes, in my opinion, is definitely a longer lifecycle is better because of a few things. First, just people who are long time in project know this project. They don’t have to learn new things. Second is communication, in agile it’s almost a

level of communication between people. If you are with a person a long time you know him, you know what you can expect from this other guy, it's easier to work together (C01007)

Similarly another member made the following comment, "I am pretty sure a longer lifecycle would be better for the teams so people already know each other, how to communicate, how to express the ideas, and the others can understand very well" (C01010). However, there may be a limitation to the number of years a member stays on the same project,

> I mean, if you are, you know, twenty years in the company but you are changing projects each two, three years I think it is okay, but if you are staying five, six, seven years on the project except from the case you are, you know, [inaudible] the summer, maybe too long from my perspective (C01005).

Although there has been rotation on the team due to people taking other jobs, layoffs, or moving to another team either voluntary or involuntary, there has not been much change and "the main meat of the team has been here since day one and I think that has been a benefit to us" (C01006). Rotation may also be due to an increase in the workload, "there were some people who left and some new people came, but it is not as frequently as like every few months or something like that, so, the rotation was not too frequent I would say" (C01010).

Specifically related to the use of an agile method and team lifecycle was the issue of estimation and velocity as mentioned by this team member:

> Also, from the perspective of making estimates if you have the same time for a long time you can better how much time can take some task to do for this team,

> because you do estimation, in fact, for the whole team right, you count velocity for the whole team, not per one person and this velocity is valid only if you have the same people in team. If you change people in team it may become invalid (C01007).

A final suggestion was that perhaps the lifecycle could relate more to tasks within the same team rather than changing teams entirely. On member said,

> that is very good when you have, you are responsible for one tasks, I don't know one month, two months, then you switch to someone else and on your place comes someone else. This is about, you know, this is something should be done because it is easier to, I don't know, for example, make some introduction for new team members or just replace some that just is sick or need to go on some vacation or just left the company, I don't know (C01009).

Member Roles

Multiple Roles within One Team. In this team roles are fairly well-defined. For the most part members do not play multiple roles within the team. On occasion when necessary there have been some exceptions, but it has not posed a problem (C01002). This was echoed in the following statement, "so I haven't seen negativity around having person play dual role or more than two, yeah" (C01010). One caution, however, was mentioned by a member of the team, "I mean we are humans so when we play multiple roles you are never giving your best for any so . . . but it kind of works out some ways" (C01008). Several of those ways are presented below.

Thus, overall members did not think that playing multiple roles within the same team negatively impacted the configuration as long as those roles are somehow related or and are not far removed from what the member does on a day-to-day basis (C01004). As one member commented,

> It depends on which roles you mean. If you mean, I don’t know, developer and guy who takes care of continuous integration, that’s fine. If you think developer and tester it is bad, not only in agile. If the roles . . . if in multiple roles are somehow related I think that is fine (C01007).

To some degree having members serve in multiple roles within the same team provides a good opportunity for cross training so that in the event a member leaves the team or is absent due to vacation or sickness another member can step in to fill that role temporarily until a new person can be hired and trained (C01010). Another member cited a similar feeling in the comment below:

> I believe that you can have different tasks . . . different role in the team, but it is very good when one, I don’t know, for example, one task can be done more than one person. This is just my own opinion, a very important thing and you know just in case it’s good to have any backup for sometimes really important, sometimes not, but just for some tasks (C01009).

From an agile methodology standpoint serving in multiple roles does not have a significant impact on project estimation as long as that member has served in those roles over an extended period:

> Yes. And, you know, if some hypothetical person in a team have even two or three roles for a long time, one of these roles, for example, is developer and

> another is CI maintainer and another is something else, doesn't matter, you know when you do estimates you know how much of work he can contribute, he or she, can contribute in a single iteration, so it doesn't matter from perspective of performance (C01007).

Additionally, members who potentially serve in multiple roles have a better understanding of what is happening in every area of the project. Ultimately, "I think that makes a more well-round person to me is a more benefit to a particular project" (C01006).

Roles within Multiple Teams. In general the team felt that members playing roles within multiple teams represented more of a hindrance than a help to the team configuration. The types of roles shared the member plays was one consideration, "you know, maybe it's depends on what role we all share. There are some roles that are difficult to connect, for instance, it's quite difficult to be project manager for instance and business analyst" (C01003). Concerns over time, split responsibilities, connectedness of the roles, and lack of "teamness" were all expressed. As one member commented,

> That always, I mean, it always make sense to me on paper that this member can be allocated to this team and this team, but it doesn't . . . typically doesn't work very smoothly, there's always some contention over that person's time or . . . (C01002).

Similarly, another member stated that they thought "it is bad because it is hard to work, you know, split the time between two projects in case when things go wrong in both, for example, right? Both projects will want you to work full-time for them" (C01007).

This member went on to say that often the concept of having members allocated for a certain percentage of time on more than one team was an illusion (C01007). Consequently another member indicated, “I think when you are pulled in all directions it is not a good thing” (C01006). This creates a negative influence on the team because a member may be very busy with one team and not have time to time to contribute to the other. Ultimately, “I think if you’re a person doing cross-functional things I think that is a negative impact when it comes to testing I believe” (C01006). The practice of crossing teams may also decrease the sense of “teamness” among the members, “if I have in the team people who are partially here and partially somewhere I do not have a team, I have a group of people who are assigned specific work” (C01003).

Although, the overall feeling that playing roles in multiple teams has a negative influence on the configuration of the team, there are potential positives if the team’s are managed properly and the member stays focused on one task at a time and does not let one team monopolize too much time. The first potential positive is the knowledge that can be gained by working in more than one team and having a more experienced member serve on multiple teams:

> Yes, very positive, especially when you have an old member who moves to a new group, that group gets to know about you and gets to learn from you from what your successes. It helps to spread what’s good happening there, so it does help a lot. You also get feedback as to what is happening on the other side, yeah, it always good as long as the person is moved to another group that is close by (C01008).

A stipulation, however, is provided by the team member. This situation is positive as long as the teams are in close proximity. Another member recounted their own experience with playing roles in two teams under the same project manager, but with a different product:

> I was able to meet different people and from their important in some organization structure and things that I was able to find out some solutions for different tasks for the other project. You know, just again gaining knowledge working on something else I was able to bring this knowledge to the team and share with them what I know, something like that. But also when you are in two teams you just need to be focused during one time on one task and thing switch to the other and sometimes it can be a little confusing to you. It can just get lost, you know.

Again, this suggestion comes with a condition. The member must be careful to keep focused on one task at a time and be careful not to get confused or overwhelmed with multiple responsibilities (C01009). Another caution is to guard against letting responsibilities on one team monopolize too much time which has the consequence of causing the other team to suffer. The member must be careful to management the splitting of time between the two teams:

> So, first if it happens and the answer is, it happens, if someone is on multiple teams I think it has a positive impact project as long as you are not spending too much time on the other team. So, let's say you are within a year, let's say you are one month within the year you are on a different project and it's split among different months . . . I mean if it is not split too much and it is not impacting the work you have to do, it has a positive effect because you, you know, you get the knowledge about interface . . . interaction with other teams . . . (C01005).

Table 64 provides a summary of the findings for virtualness or team C01:

Table 64

C01 Team Virtualness Summary of Findings

Team Virtualness	Findings
Temporal Distribution	
Time Zone	Allow for 24x7 shop, handing off work to the other team at the end of the work day; continuous, round-the-clock coverage of the project, and adds time to the work day
	Can cause scheduling problems for meetings due to few overlapping hours
	May hinder a sense of "teamness" - guard against having two teams rather than one
Use of ICT	Bridge lines for teleconferencing (primary means of daily stand-up) and individual phone calls; however, limited by time zone difference and few overlapping hours
	Desktop sharing capability of instant messaging used to facilitate agile practices such as iteration demos, iteration planning, code review, and exchanging ideas

(table continues)

Table 64 *(continued)*.

Team Virtualness	Findings
	Groupware and collaboration software such as Microsoft® NetMeeting for demonstrating iteration demos, discussing user stories, and conducting iteration planning
	Open-source application for storing backlogs and user stories
	Spreadsheet for tracking project progress
	Limited use of videoconferencing due to expense, difficult setup, availability of appropriate facilities, and intimidation related to the use of a new technology
	Limited use of drawing tablets due to difficulty in ease to use and technical problems such as inadequate band width

(table continues)

Table 64 *(continued)*.

Team Virtualness	Findings
Boundary Spanning	
Functional	Work closely with the marketing department
Organizational	Experience level between U.S. and Poland members, on an average the U.S. members had more experience than the Poland members
Cultural	There were different perspectives on working overtime initially
	Language and accents
	Word meaning, word choice, interpretation, and individual personalities
	Approach to sick leave due to differences in labor laws
	There were some differences in holiday in terms of the number of holidays and holidays celebrated over non-consecutive days

(table continues)

Table 64 *(continued)*.

Team Virtualness	Findings
Lifecycle	
Short	Keep the team from growing stale and/or becoming too complacent or comfortable
	Bringing in new members regularly refreshes and diversifies the team and generates new ideas and enthusiasm
	Shorter lifecycle allows the expertise of more seasoned and knowledgeable members to be shared across teams
	The rotation of members must be balanced, it can be detrimental to bring too many new members on to the team too quickly and vice-versa to rotate off too many experienced members too quickly

(table continues)

Table 64 *(continued)*.

Team Virtualness	Findings
Long	Finding certain skill sets in Poland can be difficult, so when people with those skill sets are hired, it is important to retain them for as long as possible
	Allows members to be very knowledgeable about the project and too focus on it specifically
	Due to the steep learning curve and the size and complexity of the project there is a necessity for a longer lifecycle for training
	Developing strong communication channels, establishing good interpersonal relations, and gaining knowledge and familiarity with project are all aspects which are developed over a longer period of time
	Related specifically to the agile practices of velocity and estimation, having the same members for a longer time the team can better assess how much time it takes to complete a task because estimation and velocity are determined for the whole team not individual members

(table continues)

Table 64 *(continued)*.

Team Virtualness	Findings
Member Roles	
Roles Within One Team	No perceived negative impact as long as roles are somehow related, i.e., serving as a developer and taking care of continuous integration
	Provides a good opportunity for cross training among members
	Estimation is not significantly impacted as long as member has been in those roles for an extended period of time
	Facilitates a more “well-rounded” member by allowing them to have a better understanding of what is happening in every area of the project

(table continues)

Table 64 *(continued)*.

Team Virtualness	Findings
Member Roles	
Roles Within Multiple Teams	Overall feeling that playing roles in multiple teams is more of a hindrance than a benefit Somewhat dependent upon types of roles and the individual Concerns over time, split responsibilities, connectedness of roles, and lack of "teamness" "An illusion" (C01007) "Being pulled in all directions is not a good thing" (C01006) Negative in that members is unable to contributed to both teams appropriately May decrease a sense of "teamness" Potential positives included: 1) knowledge that can be gained by working with other teams and 2) having experienced members serving on more than one team can distribute expertise Playing roles in multiple teams necessitates careful management

Team Agility

Extent (Degree) of Agility

For this study team agility was defined as the extent to which the team aligns with the general values and principles of agile methods as stated in the Agile Manifesto as well as with the practices of a specific method. Overall, members were not familiar the Agile Manifesto itself, but were very familiar with the general values and principles associated with agile. Values were adhered to more indirectly and informally. The team had not adopted a formal agile methodology, but had tailored multiple practices from Scrum and eXtreme Programming to meet its development needs.

As indicated by Table 63 the overall feeling was that the team adhered to all four of the values of the Agile Manifesto and provided specific examples of how each was implemented. In terms of principles, the team adhered to 11 of the 12 principles. Due to the distributed nature of the team, Principle 6 was not possible. In terms of practices the team was very agile in its development process. Multiple practices associated with Scrum and eXtreme Programming were implemented. Based upon an evaluation of these sub-dimensions of agility this team represented a high degree of agility. Table 65 briefly summarizes the team agility dimension:

Table 65

C01 Summary of Team Agility

Sub-Dimension	Comment
Values	In relation to the values of the Agile Manifesto the overall consensus of the members was that team did strive to adhere to the values, although several members indicated that they were not familiar with Agile Manifesto itself.
Principles	The team also believed that they adhered to the principles of the Agile Manifesto to a large degree and provided many examples of how the principles have been implemented.
Practices	The team had not adopted specific agile methodology, but rather had adopted individual practices from both eXtreme Programming and Scrum which best met its needs. Specific practices included: daily stand-up meetings, iteration planning, user stories, velocity, Test-Driven Development, continuous integration, common Code, standards of code, short iterations, acceptance criteria/test, iteration demos, iteration retrospectives, pair programming, automated testing, refactoring

In relation to familiarity with the Agile Manifesto, the overall consensus of the members was that the “spirit” of the Agile Manifesto was present within the team, but for the most part members were not familiar with the Agile Manifesto itself as indicated by the following member:

> I could answer yes, however, I also need to add that probably if you ask some of our team members about a specific . . . about what is inside agile manifesto some of them would not be able to answer you, because, you know, it is something what we . . . sometime ago when we start with this agile approach some of us had been more involved in all this . . . discovering this new approach, some were not and I would say the idea that are in Agile Manifesto are in our team, but not always. We are referring . . . you remember it is an Agile Manifesto, it something that is applicable to our team, but probably not all of our team member remember what is the source of this approach (C01003).

Table 66 lists the responses to the following question, "were you familiar with the values and principles of the Agile Manifesto before participating in this interview?":

Table 66

C01 Familiarity with the Agile Manifesto

Member	Comment
C01007	Honestly, I didn't know it by heart, but I was familiar with it (C01007).
C01008	Not with the Manifesto (C01008).
C01009	More or less, you know (C01009).
C01010	Not in this particular form I don't remember exact wording, but in a similar way (C01010).

Values

In relation to the values of the Agile Manifesto the overall consensus of the members was that in general there was adherence to the values within the team when asked the following question, “does the team strive to adhere to the overall values espoused in the Agile Manifesto either directly or indirectly?” as summarized in Table 67:

Table 67

C01 Responses to Adherence to Values of the Agile Manifesto

Member	Comment
C01003	Exactly. Yeah (C01003
C01006	Yeah, I believe that our team adheres to this (C01006)
C01007	Yes (C01007)
C01008	Yes, I do (C01008)
C01009	Oh, yeah, very much (C01010)

In addition to the general response to the question, several members commented specifically about how the team adhered to specific values as summarized in Table 68:

Table 68

C01 Extent (Degree) of Agility for Values

Sub-Dimension	Yes	No	Unsure	Extent of Agility
Value 1: Individuals and interactions over processes and tools	X			Although communication was cited as a challenge in a global environment, according to a member of this team the value was adhered to 100% (C01005)
Value 2: Working software over comprehensive documentation	X			General agreement to the adherence to this value (C01002, C01005, C01009, C01010)
Value 3: Customer collaboration over contract negotiation	X			The team did put a lot of emphasis on customer collaboration although it was sometime difficult due to being distributed. When actual customer was not available the business analyst would play the customer role. Contract negotiation was not applicable as the team itself was not involved in determining contractual relationships (C01002, C01003, C01005, C01007).

(table continues)

Table 68 *(continued)*.

Sub-Dimension	Yes	No	Unsure	Extent of Agility
Value 4: Responding to change over following a plan	X			Overall agreement that this value was followed with the caveat that consideration of the change would be within the context of an established procedure for change requirements (C01002, C01005, C01009, C01010)

Principles

In general the team also believed that they adhered to the principles of the Agile Manifesto. In response to the following question, “does the team strive to adhere to the twelve guiding principles outlined by the Agile Alliance either directly or indirectly?” the following responses represented the overall consensus of the members, “Okay, let me think. Yes, I believe that most of them are part of our daily work and daily approach” (C01003) and “it looks like we pretty much do the majority of those principles that are outlined” (C01006). Similarly, in response to the question, “were you familiar with the principles of the Agile Manifesto before participating in this interview?”, a representative response was made by the following team member:

> In all honesty it probably was not a structured list. The principles . . . when I went through the training this was two and a half . . . about two years ago when I went through a formal class where we were in the process of adopting the agile approach, so I was probably exposed to certain principles, but not in a twelve step list like this (C01002).

The team adhered to Principle 1 through the use of agile practices such as short iterations and small releases. Through the use of customer advocates the team had frequent communication with the customer and involved the customer during each iteration whenever possible. It was noted, however, that this principle was "balanced with company strategy" (C01008).

In relation to Principle 2 the team adhered to welcoming changing requirements "within that change request procedure that we have setup" (C01002). The team, however, did stress the evaluation of the change and how it would affect the overall project and determining just "how late" the change was being requested. Thus the team made decisions on "a case to case basis, depending on how critical the requirement is" (C01010).

Principle 3, "delivering working software frequently" was considered "a part of our every day job and we are, I think, we are quite successful" (C01003). The team had implemented two-week iterations followed by an iteration demo.

It was recognized that adherence to Principle 4 was very difficult, albeit possible, in distributed environments because business people and developers may not be in the same physical location (C01003, C01008). Frequent communication was identified as the key to implementation of this practice (C01003, C01009). The team did have business analyst working closely with developers and the quality assurance analysts (C01010) and utilized customer proxies (C01005).

Principle 5 was considered to a large degree as more implicit than explicit. One member commented, “this is something that is probably is not visible, it’s not very visible in our team, I mean not because we don’t think it is not important we just never talk about” (C01003). In terms of support, technical needs might be considered one hindrance to implementing Principle 5 (C01007). In sum, another member indicated, “yes, very much, all the individuals are pretty much motivated to practice agile and they do pretty . . . they understand the importance, the useful of the agile methodology. So, they are very, very good team, too” (C01010).

Principle 6 may be considered the most controversial principle in regard to globally distributed teams utilizing agile methods. From the responses of the members most would agree that face-to-face communication is likely the most efficient and effective means of communication. However, with a globally distributed team this is not possible and so other means of communication must be established. As discussed earlier multiple forms of ICT are used to facilitate communication, coordination, and collaboration between the team such as teleconferencing, instant messaging, desktop sharing, and email (C01002, C01008, C01010). Agile practices themselves contributed to increased communication via the daily stand-up meetings, iteration planning, iteration demos, and iteration retrospectives. Thus through the combination of ICT and agile practices the team has been able to overcome many of the challenges faced by not having face-to-face communication. It was also suggested that if possible on-site visits to each of the locations should be scheduled for selected members from the United

States portion of the team and the Poland portion of the team (C01002). Often budgetary constraints limit the amount of travel, but when feasible it appeared to have a positive influence on the team. The team also utilized customer proxies to alleviate the problem of the customer being located only in the United States (C01005). Another member suggested that specific code components should be developed in a single location (C01007).

Principle 7 was considered “something that is very important in the team and it truly works . . . I believe we clearly understand this principle, follow it” (C01003). One member mentioned that in addition to working software the team also uses user stories as a measure of progress (C01008). Agile practices such as continuous integration, unit tests, and acceptance tests were cited as beneficial for implementing this principle.

The overall consensus was that the team adhered to Principle 8. Specifically the two-week iteration was considered an appropriate time frame and was also considered sustainable over a long period of time (C01008, C01009). One issue identified that may hinder sustainability was the lack of funding or assigned hours (C01007)

In terms of Principle 9, the agile practice of refactoring helped the team to adhere to the concept of “technical excellence and good design” . As one member indicated, “we continually refactor our code when we find things can be improved, yes, and the refactoring book is always close by and we do that quite diligently” (C01008).

The team adhered to Principle 10, but as one member stated, “sometimes . . . we still get bogged down making changes for the future” (C01008). The key was to try and keep from over-engineering the application for the future. However, this did not mean that the team does not think about future functionality, as another member commented, “there is no hard and fast rule . . . it’s a judgment of the people” (C0101). It was important, however, not to get “paralyzed” by future functionality and to focus on the immediate requirements (C01008).

Principle 11 was considered very important. As one member suggested, the team must be afforded the opportunity to develop its ideas and establish a starting point for projects, “otherwise it becomes like decision by committee, and you end up with nothing” (C01008). User stories were identified as a good set of guidelines for establishing this starting point.

Finally, in regard to Principle 12 the team initially conducted iteration retrospectives, got away from it for a while, and then started up again within the past few months (C01008). This time of reflection has become a part of the review protocol which know happens on a periodic basis. As one member commented, “in the retrospective we continue to follow after the release is over so we take any negatives out of it and try to improve on that” (C01010). Table 69 summarizes the implementation of the principles of the Agile Manifesto:

Table 69

C01 Extent (Degree) of Agility for Principles

Principle #	Yes	No	Unsure
Principle 1: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	X		
Principle 2: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage	X		
Principle 3: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale	X		
Principle 4: Business people and developers must work together daily throughout the project	X		
Principle 5: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	X		
Principle 6: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation	X		
Principle 7: Working software is the primary measure of progress	X		

(table continues)

Table 69 *(continued)*.

Principle #	Yes	No	Unsure
Principle 8: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely	X		
Principle 9: Continuous attention to technical excellence and good design enhances agility	X		
Principle 10: Simplicity - the art of maximizing the amount of work not done -- is essential	X		
Principle 11: The best architectures, requirements, and designs emerge from self-organizing teams	X		
Principle 12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly	X		

Practices

According to one member in relation to practices “we are extremely agile we use a lot of the methodology” (C01004). Although the team has not adopted a specific agile methodology in its entirety, the team used a selection of practices from both eXtreme Programming and Scrum. In sum the team has tailored individual agile practices to meet its needs as suggested by the following remarks, “it is not XP nor Scrum nor nothing in this kind it is just specific mixture of practices that are suited for us. It just evolved in direction we wanted to evolve” (C01007) and “so we don’t use any names agile implementation like Scrum or something like that, we are just trying to follow some agile practices let’s say and we selected some project practices that part of our development “ (C01003). Another member indicated:

> We have picked the practices that will benefit us the most right now, unfortunately, we do not have the luxury of time, which I know is a bad excuse, so we’ve . . . the team does try to bring more practices in as the team moves forward and again, once again we’re heads down trying to develop two new features into the product, you know, full-fledged with all the teams, the team really is good about trying to do new things, improve communication, to improve the quality of the product, you know, were as we are not full-fledged, we try to pull those in every opportunity that we get (C01004).

The agile process utilized was considered informal, “I would say informal because I know from, there are a variety of teams in the organization and we’ve kind of adapted the high level agile concept to each team that makes me think we are performing fairly informally” (C01002). This member provided a helpful summary of the agile development process that the team employed:

> So we go through this every two weeks, you know, we initiate with planning, we conclude with a demo, typically project manager, myself, one of the lead developers will spend a little bit of time after the demo regrouping so we are

> ready for the next Monday morning to start that two week cycle all over again (C01002).

Another member described the high-level phases the team incorporated:

> We go from our development environment to our integration environment and then from our integration to our certification which is the landing zone right before we go to our production environment. And after we get sign off from our quality assurance then we promote our changes out to production and then we have a happy customer (C01004).

Table 70 summarizes the implementation of agile practices:

Table 70

C01 Extent (Degree) of Agility for Practices

Sub-Dimension	Description
Daily Stand-Up Meeting	Fifteen minute meeting each day via teleconference during overlapping work hours to discuss what each member has done or has planned to do for the day (C01002, C01004, C01007).
Iteration Planning	Involvement with the customer when user stories are developed and project details are discussed. Also included defining priorities, determining business criticality and complexity, and technical issues (C01003, C01004, C01007, C01009, C01010)

(table continues)

Table 70 *(continued)*.

Sub-Dimension	Description
User Stories	User stories are stored and managed via an open-source application. At the beginning of each iteration it was determined which stories would be covered during that iteration and stories were then divided up between developers and points were allocated (C01002, C01004, C01005, C01007, C01008).
Velocity	Velocity was used to determine what the team would be able to finish in an iteration. For example, if the team had 30 points and they estimated that they could finish 5 points per iteration they would then know that they need 6 iterations (C01004).
Test-Driven Development	TDD involved first writing JUnit test code and then writing the application code. Writing test upfront ensured code did what it was intended to do and limited the complexity of the code written (C01003, C01007, C01008)
Continuous Integration	Continuous integration builds were a normal part of the development process. The team utilized a device described as an orb which would turn green if the build was successful or green if it failed providing all the team members with feedback on that build (C01003, C01005, C01007, C01008, C01009).

(table continues)

Table 70 *(continued)*.

Sub-Dimension	Description
Common Code, Standards of Code, and Code Quality	The team did adhere to agreed upon standards for coding (C01003)
Simple Design	Through the use of TDD the team tried to keep the complexity of the code to a minimum because writing tests for complicated code can be difficult (C01003)
Unit Test	Implemented in conjunction with TDD (C01003)
Automated Test	Implemented in conjunction with TDD and SAHI scripts which was defined as an automation tool for building automated scripts for the deliverables (C01004)
Short Iterations	Two-week iterations (C01002, C01004, C01005, C01006, C01008, C01010
Acceptance Criteria/Test	For each story acceptance criteria and acceptance tests were written (C01003)
Iteration Retrospective	After the iteration the team discussed what they did well and what needs to be improved, and how to take the appropriate corrective measures (C01007, C01009, C01010)

(table continues)

Table 70 *(continued)*.

Sub-Dimension	Description
Iteration Demos	Demonstration of what's been completed during the two-week iteration (C01002, C01004, C01005, C01007, C01009, C01010).
Pair Programming	Used on a regular basis especially when new members joined the team to get that person involved in the development process quickly. Also used for cross-training purpose to ensure someone on the team could cover for a member who might be out due to vacation of illness (C01004, C01007, C01009)
Refactoring	Refactoring was used specifically when an existing piece of legacy code needed to be rewritten without losing that particular functionality (C01004)

Methodology

Implementation. This team implemented an agile methodology approximately three years ago and has continued to fine tune its use. The implementation of agile has had an overall positive influence on the work of the team:

> Yeah. Yep. I do and I'm a fan. I am very impressed, I think the changes have been very positive. A lot of it is the people, it kind of came with a change in regimes. But I think I think the approach is just much more conducive to

> accomplishing what we're trying to accomplish. To me it all comes down in the training when the question was ask, "what can you get done in the next two weeks" you know a lot better the answer to that question than if someone ask you "what can you get done in the next six months", and to me that's the heart of it, when you can look ahead and you know what is going to be on your desk over the next seven working days and you don't know what is going to be across your desk in a month, I think that as we've adopted it, people who are still in the organization and we have introduced it to new people who have come into the organization I think it has been a very effective . . . it has significantly enhanced our ability to develop and deliver . . . in my terms right now we are developing enhancements to an existing software application, it's a product that has been in development for about three years so we're . . . we go through a series of gathering enhancement requirements and we're applying this approach, I think it has been terrific (C01002).

Theoretically speaking one member suggested that there are some agile practices that help support a distributed environment including ". . . continuous integration, let's see, automation test and all this stuff" (C01003). As mentioned previously one of the most often cited challenges in a distributed environment is communication, as this member continued,

> It is especially difficult for agile team because the communication, face-to-face communication is very important, we do not create so much documentation that could be sent by email, we just talk and this is difficult, this is a challenge for agile team, however, from other point of view, agile offers us communication practices

like iteration planning, like stand ups, like demos, and all of this stuff supports us in being distributed. So, as I said, it is difficult to be agile, however, from other point of view there is many agile practices that have helped us (C01003)

In comparison to a waterfall methodology a member recounted their experience managing that type of process and how agile improves the development process even within a globally distributed environment:

From my past experience in managing waterfall processes and even being a developer, I can tell you that my experience from that perspective it definitely makes a team more agile, it greatly improves communication, you know, like I said between the developers and our customers and it’s a faster process, but it is also very . . . . it’s just a very . . . I can even say the word . . . efficient . . . it is more efficient and like I said by involving the customers upfront there is a lot less grey. There is a lot gray and lot less that you have to go back and redo. And if you have to redo it, it is something that you did just two weeks ago or this week or last week, it’s familiar in your mind and you know what you have to go back and change – it is not something you did three months ago and you kind of remember where you made those code changes and can’t understand how the requirements changed between then and now (C01004).

In sum, since the team implemented agile approximately three years ago, one member stated, “I can tell you I have seen improvements as the team learned it and honed it and really made it a craft. They are very good at it (C01004). In general the agile process that the team followed is summarized in Table 71:

Table 71

C01 Agile Development Process

C01 Agile Development Process
1. Begin with elaboration phase and receive initial scope from marketing and develop high-level estimate 2. Begin iteration planning to create and define high-level stories and place them in the backlog 3. Define priorities, business critical functions, and complexity based upon stories 4. Determine dependency on other applications or groups 5. Define the points 6. Conduct planning sessions every two weeks with developers and market advocates 7. Begin two-week iteration 8. Complete and commit to stories and pull from the backlog with eXPlainPMT 9. Conduct daily standup calls over two-week iteration 10. Conduct iteration demo with customer and/or customer advocate and/or external customers to show deliverable 11. Conduct iteration retrospective and define releases 12. Move unfinished stories to next iteration 13. Begin iteration planning for next two-week iteration and begin taking on additional stories (C01004; C01010).

Benefits. Even within a globally distributed environment using an agile method was identified as a benefit, “every project that we’ve had, either that I have been managing that’s been in agile so far has been a huge success” (C01004). Similar to the use of agile for ˜ollocated teams, the close customer collaboration was cited as a benefit within this team as well:

> The customers are engaged from day one and throughout the process as opposed to in waterfall where they throw, you know, some requirements over the wall at you and they see you in four months and they start testing and it is not what they expected or they are testing things you have never thought of (C01004).

The use of agile facilitates much more involvement in the process:

> the agile methodology is so much different in waterfall in that the team is collaborating everyday, whereas before you could see, you know, you would have two or three developers sitting, you know, three feet from each other and not talking to each other for a couple of weeks because they are sitting their coding and little window of opportunity and there is not a lot of cooperation and there’s a lot of things that can go wrong just because that communication is not there (C01004).

As this member indicated based upon their experience and background, “what I have seen so far in the three plus years that we have been doing agile, I would never go back to waterfall again. It just isn’t the best approach in my mind” (C01004). One way to be successful using agile is to get buy-in from all stakeholders involved, upper management, project management, team members, and the customer. However, in lieu of buy-in across the board, one of the best ways is to show that the “proof is in the pudding, you get them involved and you deliver, you strive to get the communication opening and that’s what does it, that’s what builds, you know, a solid team from developer up to your customer” (C01004).

A second benefit identified was the team’s ability to get code finished faster. Having the opportunity to see what the programmers have accomplished every two weeks via the iteration demo contributes to the speed of development, “so to me that is a benefit that things are being coded faster” (C01006). Previously it was taking the team approximately six to seven weeks to complete code.

A third benefit cited for using agile over waterfall was the fact that “it brings together a large group to move forward whether they are in one place or distributed” (C01008). Through the practice of the daily stand-up meeting the team maintains regular communication and receives up-to-date feedback which is so vital to a successful project. This daily communication was also associated with building stronger personal relationships within the team as opposed to a waterfall approach. By talking to people on a daily basis members get to know each other better and thus know one another by name and associate the task being worked on to a specific person. Communication also contributed to steering changes as members constantly reach out and say, “hey, I’ve done this change” (C01008). Additionally daily communication has led to the reduction of errors in the code.

Challenges. Challenges were also identified. One challenge that has needed to be overcome is the customers need for a written design document. Something tangible at the outset of the project that could be touched, felt, seen, to ensure progress was being made. Thus, it was considered a “challenge because there’s not something they can get their hands on so to speak and hold you accountable for and themselves accountable for which is why it is so important to have that communication with them often” (C01004). A way of alleviating this issue has been to make sure and include the customer in planning sessions and iteration demos and then when . . .

> they start seeing it every two weeks it actually gives them a sounder piece of mind because they are involved and especially when they see things that they can get their hands on and say “oh, that’s not really what I mean” here’s what I meant and then they see that it changes and the see that they are being listened to and their input is being taken seriously. I think that . . . that has really built up a better relationship in my mind (C01004).

A second challenged encountered was the fact that not initially the team’s counterpart in marketing was not following an agile process. Other application areas must begin to develop an understanding of agile in support of the software development area, “it was not an overnight thing for sure, but now that they have seen it in action, seeing it work, you know, and had very happy external customers that helps considerable” (C01004). In the organization there are still other areas which are not agile. There are still areas following a waterfall methodology, “so trying to work with them can be a little bit of a challenge to” (C01004). One way of alleviating this issue was to help them to “understand the terminology just in general, that gets them understand that and kind of what you are looking and how you want to work for” (C01004).

See Table 72 for summary of issues related to implementation of agile methodology:

Table 72

C01 Agile Methodology Implementation Summary

Issue	Comments
Implementation	Implemented approximately 3 years ago
	Heart of agile: “What can you get done in the next two weeks?”
	Agile practices that support distributed team included continuous integration and automated testing
	Communication is difficult for agile team since very little documentation is written which can be emailed, primarily just talking
	Agile practices which support communication: iteration planning, stand-up meetings, demos
	Improves communication between developers and customers facilitating a faster process over waterfall approach
	If something has to be redone it is much easier to remember what you did two weeks ago as opposed to three months ago
	Success dependent on buy-in from all stakeholders

(table continues)

Table 72 *(continued)*.

Issue	Comments
Benefits	All agile projects have been huge success
	Close customer collaboration
	Much better than the “throwing requirements over the wall” of the waterfall approach
	Daily collaboration
	“Proof in the pudding”
	Ability to code faster
	Brings together a large group to move forward together
	Successful project dependent upon receiving up-to-date feedback
	Daily communication builds stronger personal relationships
	Regular communication contributes to steering changes
Challenges	Customers wanting something tangible at the outset of the project such as a design document
	When participating functional units are not following an agile process, but rather more of waterfall approach

Cross-Case Analysis

The following cross-case analysis addresses each dimension of the theoretical framework: team structure, virtualness, and agility and puts forth propositions grounded in the data. A summary of all propositions is provided at the end of the chapter.

Team Structure

Task Design

Meaningfulness. In general the teams believed that it was important to individual members that the project they were working on held some degree of meaningfulness. “Seeing the big picture” was a common phrase that was repeated throughout the interviews. As such a part of meaningfulness was related to understanding the overall purpose and goal of the project and developing a vision for something greater than the individual components. As one member commented, it was important to avoid “getting caught down in the trenches” (C01004). Gaining a clear understanding of the purpose was also related to motivation and quality. Members desired to feel that the project had the potential to be successful. As such, in terms of the overall project purpose and goal, Yap (2005) indicated that it was important to gain “buy-in” from the team. Thus the following is proposed:

> *Proposition S1: A successful globally distributed agile team configuration emphasizes an understanding of the overall project purpose and goal and provides a vision of the “big picture” in an effort to increase meaningfulness, motivation, and quality*

A second overall concept that was prevalent involved the visibility or criticality of the project. Whether or not the project was of a critical nature or high profile to the organization appeared to afford it more meaning. Thus, new product development or the use of new technologies increased a feeling of meaningfulness over more maintenance-related projects and members considered it a privilege to work on such projects. Members needed to believe that the project would be beneficial to the stakeholders, that it would actually be used; otherwise, the meaningfulness associated with the project decreased. Thus the following is proposed:

Proposition S2: A successful globally distributed agile team configuration takes into consideration the visibility and criticality of a given project with the understanding that some projects will have a greater or lesser degree of these attributes and thus the meaningfulness to the members will be somewhat dependent upon this level

Meaningfulness was also related to the specific concepts of accountability, responsibility, trust, motivation, growth, ownership, and challenge. The overall sense was that members did not just want to show up and put in their eight hours, they wanted to make a contribution to something larger than themselves or their team. The basic idea was that by giving a member a challenging task it would show a greater level of trust, instill increased accountability, and foster stronger personal and professional growth. Interestingly, the data indicated that to some degree ownership of the project may even take precedence over importance, likening it to the relationship between a parent and a child, the project in essence becomes “like their baby” (C01007). Thus the following is proposed:

Proposition S3: A successful globally distributed agile team configuration takes the concepts of accountability, responsibility, trust, motivation, growth, ownership, and challenge into account for the purpose of increasing the likelihood of members to find meaningfulness in the project and to develop a personal interest in it

Level of experience and age were also specifically mentioned in terms of meaningfulness. The more experienced or seasoned members appeared to need a lower level of motivation, while the less experienced or new members of the team were very much interested in the profile of the project. Similar comments were made in regard to age of the members. Whereas, older members were less motivated by the meaningfulness of the project, younger members saw more visible project as a means to personal growth and promotion. Thus the following is proposed:

Proposition S4: A successful globally distributed agile team configuration realizes the differences in experience and age as determinants of the level of meaningfulness ascribed to the project by its members

Agile practices such as short iterations and small releases may also help foster an overall sense of meaningfulness as members are able to focus both on the smaller components of the project as well as envision the overall goal. As one member commented, shorter iterations are the important factor because successfully completing those small iterations leads to overall project success (C01006). The implementation of short iterations and small releases as a part of the task design may decrease complexity, provide small victories for the team, help them to keep the big picture in sight by letting them see the smaller parts being completed, and help motivation by keeping members from being overwhelmed by the size and complexity of the project. This idea also takes into consideration how the work is distributed across the team. It was suggested that two teams in two different locations should not be working on the same component or same feature. This might have the affect of attributing more meaning to that component since the portion of the team working on it could see it through from start to completion. Thus the following is proposed:

Proposition S5: A successful globally distributed agile team configuration emphasizes the use of agile practices such as short iterations and small releases to facilitate meaningfulness and motivation within the team by alleviating the overwhelmingness that may accompany complex projects and facilitate small victories

Table 73 summarizes the key words and phrases that emerged from the within-case analysis for each team in regard to meaningfulness:

Table 73

Structure - Task Design - Meaningfulness

A01	A02	B01	B02	C01
Big picture	Interesting	Balance	Criticality	Big picture
Effort	Challenging	Significance	Age	Challenging
Experience	Size	Beneficial	Experience	Importance
Visibility	Visibility	Usefulness	Promotions	Usefulness
Criticality	Distribution			Quality
Privilege				Accountability
Interest				Responsibility
Responsibility				Ownership
Pressure				Trust
Beneficial				Motivation
Success				Growth
Professionalism				Usefulness
				Distribution

Autonomy. Overall the data indicated that a high to moderate level of autonomy was necessary for the successful configuration of a globally distributed agile team. However, it was suggested that the level of autonomy must be tempered by the experience level of the individual member and the nature of the project itself. Thus, more seasoned members would typically have a higher level of autonomy than less experienced members and streamlined activities would necessitate a higher level of autonomy than critical activities (A01012). The teams also indicated that autonomy was dependent upon the management style of the project management team and organizational limitations. It was noted that being micromanaged had the potential to be detrimental to the project. If responsibilities were made clear, capable leadership was put into place, and individuals and small groups within the team were allowed to plan and manage themselves the configuration of the team would be more successful. Thus flexibility in terms of autonomy increased the possibility for productivity and engagement. It was suggested that an appropriately balanced approach to autonomy contributed to the overall "well-being" of the team (B01003). Thus the following is proposed:

Proposition S6: A successful globally distributed agile team configuration allows for a high to moderate level of autonomy dependent upon the experience level of the individual members and the nature of the project itself

Due to the globally distributed nature of the teams it was deemed important that individual members and small groups of individuals working together should be given a substantial degree of freedom and flexibility to make decisions in regard to the best approach to take on their particular task without having to run these decisions by the project manager in every situation. These decisions could include those related to business and technical issues. This was highlighted specifically in regard to offshore developers who would be left waiting for a response from the onshore project manager and thus diminishing the level of agility of the team. This was especially true if the decision did not impact the scope of the project or alter the prioritization of scheduled tasks. Thus the following is proposed:

Proposition S7: A successful globally distributed agile team configuration allows a substantial level of freedom and flexibility in terms of business and technical decisions as long as those decisions do not change the project scope or alter prioritization thus not diminishing the agility of distributed team members

Terms such as creativity, ownership, accountability, shared responsibility, collaboration, and encouragement were all associated with autonomy. Members should be given the opportunity for creative decision-making, to come up with novel solutions within the scope of the project. This creativity may lead to a greater degree of ownership and accountability to the project and the team. It was also noted that in regard to ownership and accountability within a globally distributed environment that the team must guard against one part of the team feeling a sense of ownership loss if the majority of decisions are made only at one site. A suggested remedy for addressing this situation was strong communication and splitting project ownership across distributed sites. In terms of the relationship between autonomy and shared responsibility it was suggested that a greater level of autonomy equaled an increased levels of feeling responsible. It was noted, however, that autonomy should not be taken for granted and that an atmosphere of openness and collaboration should be established within the team. Members should feel comfortable seeking input from others and encourage giving and receiving of ideas. Interestingly these are also terms commonly associated with agile methodologies. Thus, it appears that a connection could be drawn between the overall concepts associated with autonomy and the use of agile. Thus the following is proposed:

Proposition S8: A successful globally distributed agile team configuration emphasizes the shared concepts related to autonomy and agility including flexibility, creativity, ownership, accountability, shared responsibility, collaboration, and encouragement

Table 74 summarizes the key words and phrases that emerged from the within-case analysis for each team in regard to autonomy:

Table 74

Structure - Task Design - Autonomy

A01	A02	B01	B02	C01
Collaboration	Productivity	"Well-being"	Influence of	Crucial
Creativity	Engagement	Creativity	management	Prioritization
Flexibility	High to	Experience	Role of the	Scope
Leadership	moderate	High to	project lead	Experience
Management		moderate	High to	Freedom
Openness			moderate	Ownership
Freedom				Accountability
Experience				Communication
Criticality				Responsibility
High to moderate				Flexibility

Feedback. Due to the global distribution of the teams feedback was considered a crucial element to a successful agile team configuration. The overall consensus of the teams was that multiple sources of feedback are necessary ranging from upper management, project management, business or market advocates, individual team members, customers, and the work itself. As stated by one member, work is not done in a vacuum (B02003). The teams indicated that they needed various types of feedback to successfully complete a project according to its requirements and specifications and on a personal level desired to be recognized for the work completed and to hear “you’re doing a good” job from the vice-president all the way down to colleagues on the team. In general a high level of feedback both in terms of verbal and written was suggested. The more feedback the better, whether positive or negative, was considered beneficial. To some degree the data suggested that feedback from the customer was of greater importance than other sources due to the fact that it was closely tied to the requirements. Following customer feedback was input from individual team members, especially more seasoned members. There was also the suggestion that technical feedback was of greater importance than managerial feedback. Facilitating multiple streams of feedback is important to the agility of the team. Waiting among members slows the entire development process thus potentially decreasing agility. Thus the following is proposed:

Proposition S9: A successful globally distributed agile team configuration facilitates feedback from multiple sources in both verbal and written form, positive or negative, technical and managerial, with a special emphasis on the customer and individual team members to foster team agility

In addition to the people-related sources the teams indicated that feedback from the work itself was also beneficial because it provided immediate input. The use of agile practices was identified as a help in facilitating a greater degree of feedback. Agile practices such as daily stand-up meetings, user stories, iteration demos, iteration retrospectives, daily deliver of code to the repository (e.g., small releases, continuous integration, daily builds) and testing contributed to early detection of potential problems and enabled quicker turnaround of code. Thus the following is proposed:

Proposition S10: A successful globally distributed agile team configuration implements appropriate agile practices to glean feedback from the work itself and to facilitate a greater degree of input potentially increasing the agility of the team

Table 75 summarizes the key words and phrases that emerged from the within-case analysis for each team in regard to feedback:

Table 75

Structure - Task Design - Feedback

A01	A02	B01	B02	C01
Multi-source	Multi-source	Multi-source	Multi-source	Multi-source
Positive or negative	Technical over managerial	Understanding customer’s need	Recognition	Creation of commitment
Daily and weekly meetings	User stories	Customer over peers	Work is not done in a vacuum	Benefit of agile practices
High level		Collaboration	Verbal and written	“Doing a good job”
Benefit of agile practices		Daily deliver of code		
“Be upfront, blunt, and candid”		Quicker turnaround		
Work itself provides immediate feedback		Early detection		
		Checkpoints		
		Gauging progress		
		Creating a sense of meaningfulness		
		Work itself		

Core Norms. In evaluating the responses in regard to core norms it was somewhat difficult to distinguish between primary and secondary norms as defined by Hackman (2002). Therefore this section simply analyzes core norms as a whole without designation as primary or secondary.

Although organizational policies and procedures were in place within each team to serve as general guidelines for what should and should not be done in terms of conduct norms appeared to be developed more by the teams themselves in what was described as an evolutionary or natural way. This process or approach took place over time as members interacted with each other and developed professional and personal relationships. It was suggested that longer team duration might benefit the development of norms as members would have more time together. Due to the professionalism of the members it was not considered necessary in most teams to have “official” norms (C01010) or to construct a list of “Ten Commandments of Behavior” (A01012). Another emerging theme was that of flexibility and consistency. Overall the general feeling was that core norms should not be so strictly enforced that it restricted the work of the team. Norms should be flexible and consistent and serve the needs of the team not the other way around. Depending on the size of the team the analogy of a “class” or “family” was provided as a way to implement and practice the norms. Thus the following is proposed:

Proposition S11: A successful globally distributed agile team configuration allows for the development of core norms of conduct in an evolutionary and natural way that allows for flexibility and trust in the professionalism of its members

Whether specifically written down or communicated verbally it was suggested that norms be agreed upon at the start-up of the project and that expectations be clearly defined and communicated. Due to the globally distributed nature of the teams it was considered important that all members "be on the same page". As a part of this definition it was noted that understanding the chain of command, the corporate culture, and overall organizational policies and procedures was important. In addition, the global nature of the teams may lessen the evidence of core norms due to the lack of physical presence. It may be more difficult to gauge how members are "conducting themselves" when they cannot be seen. Thus it was suggested that a way of overseeing the members was to have a central contact person located at each site to handle any issues and strong communication channels in place. Also related to the global distribution of the team was the inclusion of members from various cultures speaking different languages. Consequently, it was noted that core norms should be accommodating of these differences. Thus the following is proposed:

Proposition S12: A successful globally distributed agile team configuration clearly defines the norms upfront either explicitly or implicitly and establishes expectation while providing accommodations for lack of physical presence and crossing cultural boundaries

In terms of the use of agile practices and the establishment of norms it was suggested that the team implement specific standards for daily stand-up meetings, iteration planning, and iteration demos. For example, daily stand-up meetings should not exceed fifteen minutes and should be centered on technical aspects of the projects with follow-up meetings between members afterwards if necessary. Due to the global distribution of the team the daily stand-up meeting may become a significant means for monitoring work progress and thus should focus on that aspect of the project. However, for iteration planning and iteration demos there should not be such strict time limitations. Thus, core norms are established which deal with technical issues such as meetings, reporting, code check-in, walk-throughs, and the actual building of software. In sum, core norms may contribute to the self-organization of the team in terms of personnel, technical, and performance considerations. Thus the following is proposed:

> *Proposition S13: A successful globally distributed agile team configuration establishes norms not only for personnel related issues but also for technical and performance related issues raised by the use of agile practices and global distribution*

Table 76 summarizes the key words and phrases that emerged from the within-case analysis for each team in regard to core norms:

Table 76

Structure - Core Norms

A01	A02	B01	B02	C01
Not project specific Confidentiality Reporting Corporate culture Professionalism Experience Flexibility Established at project start-up “Be on the same page” Physical presence Cultural boundaries Clear expectations Relationship to agile practices	Longer team duration Natural evolution	Understanding the chain of command Strong communication channels Development Within team Flexibility Consistency	Upfront expectations	Evolutionary approach Not driven by organizational policies and procedures Size Central contact person at each site Flexibility Resolution within team Overall openness Relationship to agile practices

Team Composition

Size. Overall the teams indicated that a smaller team size was more beneficial for a successful globally distributed agile team configuration. Those that argued that a larger team could also contribute to a successful configuration conceded that ultimately the team would need to be broken down into sub-teams which in essence created multiple teams each with a smaller number of members. Several criteria for determining size were suggested such as “well-being” of the team, size and complexity of the project, number of projects the team would be supporting, and budget considerations. It was suggested that a globally distributed agile team be no more than twenty members and anything larger than that be broken into sub-teams.

A common thread for supporting the configuration of smaller teams revolved around communication, coordination, and control. With fewer “contact points” communication becomes more effective thus decreasing the potential for misunderstandings, misinterpretation, and conflict. It was suggested that smaller teams were easier to manage thus increasing the ability to keep members moving in the same direction and on the “same page”. There was indication that smaller teams facilitated stronger relationship building, increased the sense of shared responsibility and “teamness”, and alleviated cultural differences. In addition to the arguments for supporting smaller teams the data indicated multiple reasons arguing against larger teams. Larger teams were identified as having more problems and issues. These included increased difficulty of managing and keeping members on the same page, decreased feedback and a tendency for members to fade into the background thus diminishing teamwork, decreased sense of shared responsibility and “teamness”, increased potential for misunderstandings and conflicts, and decreased speed at which the team could make progress. The argument was made that simply adding more members to the team would neither increase its productivity nor its agility. Thus the following is proposed:

Proposition S14: A successful globally distributed agile team configuration consists of small teams, of no more than twenty members, in an effort to foster effective communication, coordination, and control and to alleviate cultural differences within a global distributed environment

It was also noted that the possibility of configuring smaller teams was dependent upon the members being allocated 100% to the team and that the team be staffed with the necessary skill sets during project start-up. Following these criteria may help to alleviate feelings of being overwhelmed. The team should also have the flexibility to ramp up when necessary. Thus the following is proposed:

Proposition S15: A successful globally distributed agile team configuration strives to allocate members 100% to the team and to assign the appropriate skill sets at project initiation with the option of ramping up the team up as necessary

A smaller team was also identified as more beneficial for a successful configuration as the size of the team had an impact on the implementation of certain agile practices. Practices such as the daily stand-up meeting and iteration planning began to lose their focus as the size of the team grew. Overall there was less participation and interest during the meetings. If team becomes too large there was a necessity to modify the practices by conducting multiple stand-ups and iteration planning sessions with sub-teams. Additionally, if the team grew too large, the user stories were divvied up too much among members increasing the potential that members were waiting for someone else to finish their story before they could move on, in effect decreasing the agility of the team. Thus the following is proposed:

> *Proposition S16: A successful globally distributed agile team configuration consists of a smaller number of members to better facilitate certain agile practices such as the daily stand-up meeting, iteration planning, iteration demos, iteration retrospectives, and user stories*

Table 77 and Table 78 summarize the key words and phrases that emerged from the within-case analysis for each team in regard to size for small and large teams respectively:

Table 77

Structure - Composition - Size - Small

A01	A02	B01	B02	C01
Ease of management Effective communication Decrease conflict Greater sense of “teamness” Alleviate cultural differences Facilitate relationship building Possess skill sets needed Flexibility to ramp up	Small, do not add bodies just to add bodies	Emphasize good communication, feedback, and review Benefit of agile Size in relation to well-being of team	Based upon project size and number of projects supported Establish a formula for number of onshore leads and offshore developers Budget consideration	Complexity and project size Helps with estimations Agile, no more than 20 May lead to feelings of being overwhelmed Alleviate cultural differences

Table 78

Structure - Composition - Size - Large

A01	A02	B01	B02	C01
Management challenge Less feedback Tendency to fade into the background Teamwork my diminish Less of a feeling of responsibility Increased potential for conflicts Speed at which team can progress may slow down		More challenges	Can become cumbersome Must be broken into smaller sub-teams each with a project manager	Run into more problems and issues May slow down the progress Necessitates changes to agile practices Communication collaboration, and coordination more difficult

Mix. In general the teams indicated that diversity among team member was important to a successful configuration. The fostering of an open, honest, respectful, collaborative environment in which individuals of differing knowledge, experience, skill set, age, ethnicity, nationality, gender, and personality could work together to complete their work was considered a “big plus” (B02001). In addition a balanced mixture of personal, business, managerial, and technical attributes of the membership was considered essential. Many benefits were associated with this balanced mix of members including: creativity, generation of new ideas and approaches, sharing of knowledge, offering of different viewpoints and thought patterns, increased collaboration, improved innovativeness, and bringing together new and existing members. However, it was noted that too many differences may lead to miscommunication or project slow down. Thus the following is proposed:

Proposition S17: A successful globally distributed agile team configuration consists of a balanced mixture of members in terms of personal, managerial, and technical skills to increase the degree of creativity, idea generation, knowledge sharing, innovativeness, and “teamness”

Although multiple characteristics were identified there did appear to be an emphasis on business and technical-related attributes. Diversity of managerial and technical experience was indicated as a means to better understand the client’s expectations and to meet those expectations. The primary criteria for selection of technical members included skill set, need, availability, merit, location, and duration of the project. Selection criteria for the project management team consisted of expertise and appropriate domain knowledge. One of the benefits of global distribution was the ability to select members from a large talent pool and to find specialized expertise without the constraints of geographic location. Thus the following is proposed:

Proposition S18: A successful globally distributed agile team configuration includes members with the necessary business, managerial, and technical skills necessary to complete a given project and should take advantage of the large resource pool afforded by a globally distributed environment

Interestingly, many of the benefits identified in regard to mix were related to the hallmarks of agile methodologies such as creativity, idea generation and problem solving, collaboration, innovativeness and technical expertise. Thus the following is proposed:

Proposition S19: A successful globally distributed agile team configuration takes advantage of the benefits of a balanced team membership and the values and principles associated with agile methodologies

Table 79 summarizes the key words and phrases that emerged from the within-case analysis for each team in regard to mix:

Table 79

Structure - Composition - Mix

A01	A02	B01	B02	C01
More diversity the better Open, honest, collaborative culture Improves innovativeness Diversity in experience Mentoring relationships and knowledge transfer Clean division of labor Selection criteria for technical and management roles	Based upon skill set, experience, need, merit availability, and knowledge Steep learning curve Interest and specialization	Mutual Respect Balance Opportunities for ideas, creativity, and collaboration Potential for to miscommunication or project slow downs Selection based upon type of need, technical skills, duration of project	“Big plus” Kicking ideas around Coming up with new approaches Different view points, thought patterns, and how things should be done Contract members bring in experience from working with other organizations	Diversity welcomed Creativity, generation of ideas, and shared knowledge Mesh new and existing members

Knowledge and Task Related Skills. It was a given among the teams that the knowledge and task-related skills of the members were crucial to a successful configuration and that members should be continually challenged to improve those skills. It was suggested that members having the appropriate skills could actually build trust and a stronger sense of “teamness”. In regard to these types of skills it was suggested that the globally distributed nature of the team provided a large talent pool and the ability to find specialized skills without the constraint of geographic distance. The members agreed that various skill sets where necessary within the team because of the multiple roles and activities which were present. The nature of the project itself could also have an influence on the types of knowledge and task-related skills needed. Undergirding the suggestions was the idea of balance. The team needs a good balance of members with appropriate the level of managerial, business, and technical skills. Of course technical skills were pointed out consistently such as knowing how to code in a particular language, but it was also emphasized that project management and organizational skills were also crucial. Another emerging theme was that every member did not need to be an expert in every area of the project. The whole idea of a team was that each individual member brings a different set of skills, expertise, and knowledge and that the key was to exploit the strengths of each member and allow members to teach and mentor each other. Thus the following is proposed:

Proposition S20: A successful globally distributed agile team configuration includes a wide range of knowledge and task-related skills in the areas of business, project management, and technical necessary to complete projects and takes advantage of the global nature of the team by drawing from a large pool of specialized resources

It was widely acknowledged and accepted that compensation among the members did take place. Due to the hiring of new members, project complexity, or a steep learning curve it was not unusual for members to help out others who were not as strong in a given area. Of course it was indicated that having all the required skills present from the outset of the project was beneficial it was not always feasible. Due to financial constraints it was not always possible to hire experts in every area, thus members needed to continue to learn and hone their skill sets. However, it was noted that the team needs to practice caution that members do not develop a lackadaisical attitude while relying on others to help them or that the compensation become a detriment to the team. One of the major jobs of the project manager was identified as monitoring the inner workings of the team and making sure each member was carrying their appropriate load. Thus the following is proposed:

Proposition S21: A successful globally distributed agile team configuration acknowledges and accepts the need for compensation and strives to assist members who are not as strong in a particular area until they reach the necessary level of expertise

It was also suggested that the use of an agile methodology can actually contribute to getting new members up to speed quicker. Teamwork, collaboration, and strong technical skills are hallmarks of agile methodologies so it was important to call upon the expertise of existing members to assist new or less experienced members. For example, the use of pair programming allowed less experienced members to team up with a more seasoned members and work together until the less experienced members were comfortable with how things in the team worked. Thus the following is proposed:

> *Proposition S22: A successful globally distributed agile team configuration takes advantage of agile practices which promote teamwork and help new or less experienced members to increase their knowledge and task-related skills*

Table 80 summarizes the key words and phrases that emerged from the within-case analysis for each team in regard to knowledge and task-related skills:

Table 80

Structure - Composition - Knowledge and Task-Related Skills

A01	A02	B01	B02	C01
Different skill sets needed Assignments based upon value Nature of the project itself and team role Required skills from the outset Trust and “teamness” Balance Complexity of learning process Necessity of business skills Financial constraints	Differing levels of experience Include junior and senior members	Adequate technical skills to decrease additional training Must have solid technical experience and know how to code Compensation is possible at times, but one person cannot do the work of two people for an extended length of time	Every member does not need to be an expert in all areas Exploit member’s strengths Broad range of skills necessary Knowledge transfer and capturing Development of mentoring and teaching relationships	Contribution of agile Learning curve Challenge members Naturally talented members teach and mentor other members Every member does not to be an expert in all areas Motivation is most important

Interpersonal Skills. There was overall agreement that interpersonal skills are very important even within globally distributed teams. Perhaps more so than in colocated teams since it was suggested that global distribution makes it easier to leave members out of the communication loop and that members may more easily fade into the background. Thus interpersonal skills were identified as the key to help in overcoming challenges such as communication, culture, and language. As one member commented, the team cannot work in a vacuum (A02001). It was noted that there were multiple benefits to members possessing strong interpersonal skills including: communicating more effectively, expressing ideas more clearly, generating ideas, working together to solve problems, building mutual respect, cooperating, developing a common vision, helping members feel more comfortable in expressing themselves, decreasing the need for written documentation, increasing reliance on informal communication, building a stronger sense of "teamness", and keeping an open-mind to differences. Thus the following is proposed:

Proposition S23: A successful globally distributed agile team configuration seeks members with strong interpersonal skills and encourages the development of these skills in order to take advantage of the many benefits they provide in a globally distributed environment

Additionally, the use of agile methods may even increase the importance of interpersonal skills due to the fact that agile relies heavily on daily interaction between members. Agile is predicated on the concepts of communication, interactions, people, and collaboration, thus It was noted that strong interpersonal skills contributed to such agile practices as the daily stand-up meeting which was conducted via ICT rather than face-to-face. Because agile practices involved daily interaction it was suggested that an agile methodology can actually help to foster stronger interpersonal relationships. Thus the following is proposed:

Proposition S24: A successful globally distributed agile team configuration takes advantage of the agile practices which emphasize daily interaction and utilize these practices to foster stronger interpersonal relationships

In spite of the many benefits provided by the possession of strong interpersonal skills among team members, it was recognized that projects can be completed even in the absence of these types of skills. Although interpersonal skills are important and beneficial, these types of skills were not considered absolutely necessary. However, it was noted that when interpersonal skills were lacking delays occurred and members potentially got side-tracked with personal conflicts. Even when interpersonal conflicts arose, the teams indicated that members should still respond in a professional manner and not their personal differences interfere with the work. Thus the following is proposed:

Proposition S25: A successful globally distributed agile team configuration acknowledges that projects can still be completed in the absence of strong interpersonal skills but that personal conflicts may cause delays and all differences should be handled in a professional manner

Table 81 summarizes the key words and phrases that emerged from the within-case analysis for each team in regard to interpersonal skills:

Table 81

Structure - Composition - Interpersonal Skills

A01	A02	B01	B02	C01
Communicate effectively Delays in the project Potential for more reliance on informal communication Less need for written documentation Stronger sense of "teamness" Challenges of global distribution Important but not absolutely required	Cannot work in a vacuum Must talk and generate ideas Work together to solve problems Open-mindedness to differences	Develop mutual respect Establish regular communication Personality conflicts can halt progress Cooperation Vision for the team	Important, but project can still be completed Respond professionally	Agile is predicated on the concepts of communication, interactions, people, and collaboration Benefit of agile practices Challenges of global distribution

Team Virtualness

Temporal Distribution

It was suggested that geographic distance was not nearly as troublesome as time zone differences. Although benefits were identified such as 24x7 coverage of the project, handing off work, and adding time to the day, the overall feeling was that having some degree of overlap in work hours between sites was very important to a successful configuration. Limited overlap in work hours hindered access to members, decreased timely communication, lessened a sense of "teamness", and caused difficulty in scheduling of meetings. All of these challenges presented a problem in terms of agile since it emphasizes on-going collaboration, regular communication, and daily stand-up meetings. Thus the following is proposed:

> *Proposition V1: A successful globally distributed agile team configuration provides at least some degree of overlapping work hours between sites to facilitate the use of agile values, principles, and practices*

All the teams participating in this study utilized some measure of information and communication technologies. Overall, it was agreed that teleconferencing, instant messaging, desktop sharing, and collaboration software were essential in regard to team configuration in relation to temporal distribution. Specifically teleconferencing and instant messaging were primary means of conducting daily-standup meetings, iteration planning, iteration demos, and iteration retrospectives. Desktop sharing was used to facilitate iteration demos, iteration planning, code review, and idea exchange. The teams reported limited use of videoconferencing due to difficulty in ease of use, complicated setup, and inadequate band-width. Thus the following is proposed:

Proposition V2: A successful globally distributed agile team configuration makes use of multiple communication channels through the use of teleconferencing, instant messaging, desktop sharing, groupware, and email technologies

Tables 82-83 summarize the key words and phrases that emerged from the within-case analysis for each team in regard to temporal distribution:

Table 82

Virtualness - Temporal Distribution - Time Zone Differences

A01	A02	B01	B02	C01
Limit instant access to members Communicating in a timely fashion Scheduling meeting times Should be some overlapping hours Geographic distance not a major problem	Difficulty in scheduling times for meetings Need for overlapping work hours Daylight savings time should be taken into consideration	Difficult when no overlapping office hours between locations	Difficult when no overlapping office hours between locations	Allow for 24x7 shop Can cause scheduling problems for meetings May hinder a sense of “teamnness”

Table 83

Virtualness - Temporal Distribution - Use of ICT

A01	A02	B01	B02	C01
Telephone, desktop sharing, whiteboard, email, Jabber IM to decrease phone expense and for scheduling meetings Collaboration and desktop sharing applications	Teleconferencing and instant messaging for synchronous communication Groupware and collaborative software for desktop sharing Shared code and requirements repository Digital photo album	Email and teleconferencing	Phone and email No chat technologies Visual communication via Microsoft® Visio	Teleconference Desktop sharing Instant messaging Groupware and collaboration Open-source application for storing backlogs and user stories Spreadsheet for tracking project progress

Boundary Spanning

Functional. Three teams did not identify any specific functional boundaries. Of the other two teams, one indicated that perhaps functional boundaries were noticeable in larger organizations. As was mentioned product delivery, in this case software, involved different functional teams. It was also suggested that when multiple functional units were involved it was important to establish clear divisions of labor. The second team indicated that it worked closely with the marketing department on its projects. The benefit to this team was that the marketing department had also implemented an agile process which made crossing this boundary much smoother. In the event that other functional units were participating using a more waterfall approach there was more difficulty.

Organizational. The two teams from Organization B indicated that the team crossed organizational boundaries in that they had members from an offshore contracting company. Thus offshore members were accountable both to the contracting company to which they were employed and to the project management team of Organization B. Due to this interaction between two separate organizations it was considered important that standards and guidelines be established upfront in regard to the chain of command and reporting mechanisms. Additionally, it was important to understand how offshore members were assigned to the team. This was echoed by team A02 which suggested that using the same guidelines, working patterns, and best practices helped to overcome these types of boundaries. A consistent theme which emerged was related to the reporting practices. A final organizational boundary mentioned related to the difference in experience between the members in team C01. Overall, the teams did not believe that crossing organizational boundaries presented a problem for a successful configuration.

Cultural. Overall the teams cited differences in holidays, language issues (e.g., accents, word choice and meaning, interpretation), views about overtime and sick leave as cultural boundaries encountered. But the overall consensus was that crossing these cultural boundaries was not a significant deterrent to a successful team configuration. As one member indicated, the “world is getting smaller” (A01006). Thus in terms of boundary spanning the following is proposed:

> *Proposition V3: A successful globally distributed agile team configuration may cross multiple boundaries in regard to functional, organizational, and cultural without negative affect, but organizational “buy-in” and the use of an agile process by other functional unites are beneficial*

Tables 84-86 summarize the key words and phrases that emerged from the within-case analysis for each team in regard to boundary spanning:

Table 84

Virtualness - Boundary Spanning - Functional

A01	A02	B01	B02	C01
Product delivery involves different functional teams More noticeable in large companies Clear division of labor "Productive" versus "unproductive" overlap	None identified	None identified	None identified	Work closely with the marketing department

Table 85

Virtualness - Boundary Spanning - Organizational

A01	A02	B01	B02	C01
"Job families" may have different reporting structures Time-tracking system used organizational wide is a bit inefficient	Using same guidelines, working patterns, best practices U.S. team members and Brazilian team members report via separate hierarchies	Offshore members are accountable to both contracting company management and onshore organization management	Initial resistance to offshoring Seniority based approach to assigning onshore lead	Experience level between U.S. and Poland members, on average the U.S. members had more experience than the Poland members

Table 86

Virtualness - Boundary Spanning - Cultural

A01	A02	B01	B02	C01
Much less of a boundary since the “world is getting smaller” Tools for calibrating to different cultures Involving offshore member when scheduling meetings	Differences in holidays and vacations Language barrier	Differences in holidays	Language differences and mutual understanding	Views on overtime- Language issues Approach to sick leave Holiday differences

Lifecycle

In terms of lifecycle there was a mixed response among the teams and the individual members. Arguments were made that both a short lifecycle and a long lifecycle was the most beneficial for a successful configuration. This section will analyze arguments for each and suggest an overall conclusion.

The following arguments were made in support of a shorter team lifecycle. First, a shorter lifecycle actually allows the team to see their work put into production. Second, it was suggested that a shorter lifecycle might keep the team from growing stale, becoming too complacent, or getting too comfortable. It was stated that bringing new members on a regular basis would refresh and diversify the team and foster the generation of new ideas and a sense of enthusiasm. Additionally, a shorter lifecycle would allow more experienced and knowledgeable individuals to move from team to team sharing their expertise. Regardless, it was noted that rotating members should be done in a balanced member. Moving too many members off the team at one time or putting too many members on the team at one time could have a negative impact on the configuration.

Those advocating a longer lifecycle pointed to the following arguments. Keeping the team together for a longer time allowed for developing of stronger professional and personal relationships, creating more stability, breeding of familiarity with the project and the strengths of individual members, establishing consistency in work processes, contributing to increased buy-in and commitment, retaining hard to find skill sets, increasing knowledge, adjusting to a steep learning, size, and complexity of the project, contributing to better communication and collaboration, avoiding the need to ramp up, developing strong communication channels, utilizing agile practices such as velocity and estimation. In sum it appeared that teams preferred a longer lifecycle over a shorter lifecycle, although arguments were made in support of both. Thus the following is proposed:

> *Proposition V4: A successful globally distributed agile team configuration consists of a longer team lifecycle to allow for the development of strong personal and professional relationships; creation of stability, familiarity, consistency, and commitment; retention of hard to find skill sets; adjustment to steep learning curve, size, and complexity of projects; and avoidance of the need to ramp the team up*

Tables 87-88 summarize the key words and phrases that emerged from the within-case analysis for each team in regard to lifecycle:

Table 87

Virtualness - Lifecycle - Short

A01	A02	B01	B02	C01
Some members felt that with good dynamics and good communication it should not matter	Opportunity to see work implemented Progressive lifecycle, bring members on over time	N/A	N/A	Keep the team from growing stale and/or becoming too complacent or comfortable Bringing in new members regularly refreshes and diversifies the team and generates new ideas and enthusiasm Shorter lifecycle allows the expertise of more seasoned and knowledgeable members to be shared across teams Calls for balanced rotation of members

Table 88

Virtualness - Lifecycle - Long

A01	A02	B01	B02	C01
Creates a more stable team Develops stronger personal relationships Breeds familiarity Contributes to greater buy-in and more committed members	Allows time for members to get to know each other better and develop personal relationships which may contribute to better communication	Build consistency Avoid ramping up with new members on a regular basis	May be more dependent on quality of person rather than time More beneficial for large systems with a lot of maintenance Retaining the same offshore members year to year	Finding certain skill sets may be difficult, Increases knowledgeable Steep learning curve, size, and complexity Establishing interpersonal relations Gaining knowledge and familiarity Agile practices of velocity and estimation

Member Roles

Multiple Roles within One Team. The general feeling of the teams was that members playing multiple roles in the same team did happen to some degree on most teams and that it did not negatively affect a successful configuration. The teams indicated that members playing multiple roles within the team could actually be beneficial if the roles were managed responsibly and the roles were somehow related (e.g., developer and overseeing continuous integration). Several benefits were identified including the following: contributing to personal growth and care development, enabling members to expand their horizons and understanding of the project, alleviating boredom, keeping the project more interesting, teaching prioritization skills, instilling appreciation for what other members do on the team, providing opportunities for cross-training, and developing more well-rounded members. It was emphasized that the team needs to be careful not to overload the members with too many roles and that strict adherence to the role structure might actually be counter-productive. Members playing multiple roles within the team did not appear to impact the use of agile practices such as estimation as long as member had been in particularly roles for an extended period. Thus the following is proposed:

Proposition V5: A successful globally distributed agile team configuration allows members to play multiple roles within the same team in an effort to take advantages of the multiple benefits it affords

Roles Within Multiple Teams. There was mixed response in regard to members playing roles within multiple teams. It was indicated that to some degree whether or not members should play multiple roles within multiple teams was dependent upon the roles, number of roles, the individual, the time allocated to each team, and the overall size of the team. Several positives were identified for members crossing teams including: filling in for members out sick or on vacation, gaining knowledge by working with other teams, and having experienced members serving on more than one team to distribute expertise. Despite these potential benefits the overall feel was that it was more of a hindrance than a benefit and could potentially lead to a negative impact on the team configuration. As one member commented, it's “an illusion” (C01007).

Multiple reasons were provided for the negative impact of members playing roles in multiple teams to a successful configuration. The first emerging theme involved commitment and contribution. When members are split across teams it was suggested that it makes it difficult for teams to develop a sense of commitment and to make an appropriate contribution to each team. As such, members were pulled in all directions and as one member commented, “being pulled in all directions is not a good thing” (C01006). In the end it was indicated that it was difficult for any of the teams to get their fair share of member's allocation with a remote, partial resources. It was even suggested that playing roles in multiple teams would decrease a sense of “teamness” as responsibilities are split and less time was spent with any one team. A second theme was the potential conflict created between project managers in regard to who has priority of member's time creating issues of who gets the member first. A related issue involved conflicts in regard to delivery schedules and expectations. In the end the overall result might be that productivity may suffer. Thus the following is proposed:

Proposition V6: A successful globally distributed agile team configuration primarily consists of full-time members allocated 100% to one team as opposed to members crossing multiple teams

Tables 89-90 summarize the key words and phrases that emerged from the within-case analysis for each team in regard to member roles:

Table 89

Virtualness - Member Roles - Within One Team

A01	A02	B01	B02	C01
Contributes to the growth and career development May actually be a positive as members recognize that playing multiple roles enables them to expand their horizons	Dependent upon individual Involve all members in some way throughout the project Strict adherence to roles can be counter-productive	Must be careful not to overload resources	Beneficial as long as roles are managed responsibly May alleviate boredom, keep project more interesting, teach prioritization skills	No perceived negative impact as long as roles are somehow related Provides a good opportunity for cross training Estimation is not significantly impacted

Table 90

Virtualness - Member Roles - Within Multiple Teams

A01	A02	B01	B02	C01
Dependent on individuals and specific roles, number of roles, overall size of the team Members cannot be fully committed Members pulled in all directions Productivity may suffer May create conflicts between project managers	Dependent upon roles played and individual	Dependent upon amount of time that must be allocated May lead to overtime May create conflicts between project managers Creates issue of who gets the member first Beneficial when members are out sick or on vacation	Dependent upon the individual May cause conflicts between which team's project takes priority	Somewhat dependent upon types of roles and the individual Concerns over time, split responsibilities, connectedness of roles, and lack of "teamness" Members are unable to contribute to both teams appropriately

Team Agility

In terms of agility which was based upon the extent to which the teams adhered to the values and principles of the Agile Manifesto as well as the practices of a specific agile methodology the cross-case analysis indicated the following. The responses to the questions, “Does the team strive to adhere to the overall values espoused in the Agile Manifesto either directly or indirectly?” and “Does the team strive to adhere to the twelve guiding principles outlined by the Agile Alliance either directly or indirectly?” most likely stem from the fact that in general the team members were not familiar with the Agile Manifesto, although there were some exceptions. Although the overall feeling of each team was that they adhered to the values and principle indirectly, an analysis of the interviews suggested that in some ways the teams did follow certain values and principles more directly and provided specific ways that each was implemented. Table 91 summarizes the responses:

Table 91

Overall Familiarity with the Agile Manifesto and Adherence to Values and Principles

Question	A01	A02	B01	B02	C01
Were you familiar with Agile Manifesto before seeing it for this interview?	N	N	N	N	N
Does the team strive to adhere to the overall values espoused in the Agile Manifesto either directly or indirectly?	I	I	I	I	I
Does the team strive to adhere to the twelve guiding principles outlined by the Agile Alliance either directly or indirectly?	I	I	I	I	I

Note: Y = Yes; N = No; U = Unsure; D = Directly; I = Indirectly.

Values

An analysis of the responses of each team in regard to the extent of agility for adhering to the values of the Agile Manifesto suggested that these values can be successfully implemented in a globally distributed environment. See Table 92 for a summary of the responses:

Table 92

Overall Adherence to Values and Principles of the Agile Manifesto by Team

Value	A01	A02	B01	B02	C01
1 - Individuals and interactions over processes and tools	U	Y	Y	Y	Y
2 - Working software over comprehensive documentation	N	Y	Y	Y	Y
3 - Customer collaboration over contract negotiation	N	Y	Y	Y	Y
4 - Responding to change over following a plan	U	Y	Y	Y	Y

Note: Y = Yes; N = No; U = Unsure.

Principles

In analysis of the responses of each team in regard to the extent of agility for adhering to the principles of the Agile Manifesto indicated that with the obvious exception of principle 6 it was possible to successfully implement the other principles in a globally distributed environment. See Table 93 for a summary of the responses:

Table 93

Cross-Case Analysis Extent (Degree) of Agility for Principles

Principle	A01	A02	B01	B02	C01
1 - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software	Y	Y	Y	NC	Y
2 - Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage	Y	Y	Y	NC	Y
3 - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale	U	Y	Y	NC	Y
4 - Business people and developers must work together daily throughout the project	Y	Y	Y	NC	Y
5 - Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done	Y	Y	Y	NC	Y
6 - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation	N	N	N	NC	N
7 - Working software is the primary measure of progress	U	Y	U	NC	Y

(table continues)

Table 93 *(continued)*.

Principle	A01	A02	B01	B02	C01
8 - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely	U	Y	Y	NC	Y
9 - Continuous attention to technical excellence and good design enhances agility	U	Y	Y	NC	Y
10 - Simplicity - the art of maximizing the amount of work not done -- is essential	Y	Y	NC	NC	Y
11 - The best architectures, requirements, and designs emerge from self-organizing teams	Y	Y	Y	NC	Y
12 - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly	U	Y	Y	NC	Y

Note: Y = Yes; N = No; U = Unsure; NC = No Comment.

Thus the following is proposed:

> *Proposition A1: A successful globally distributed agile team configuration adheres to the "spirit" of the values of the Agile Manifesto either directly or indirectly although individual members may not be familiar with the Agile Manifesto in its official form*

Proposition A2: A successful globally distributed agile team configuration adheres to the "spirit" of the principles of the Agile Manifesto either directly or indirectly although individual members may not be familiar with the Agile Manifesto in its official form

Practices

An analysis of the responses of each team in regard to agile practices within a globally distributed environment indicated that multiple practices can be successfully implemented in a globally distributed environment. Thus the following is proposed:

Proposition A3: A successful globally distributed agile team configuration Implements multiple agile practices through tailoring the practices to its specific needs and uses ICT to effectively overcome the lack of face-to-face interaction through synchronous and asynchronous communication

See Table 94 for a summary of the responses:

Table 94

Cross-Case Analysis Extent (Degree) of Agility for Practices

Practice	A01	A02	B01	B02	C01
Acceptance Criteria/Acceptance Testing					X
Automated Testing		X			X
Common Code / Standards of Code / Code Quality					X

(table continues)

Table 94 *(continued)*.

Practice	A01	A02	B01	B02	C01
Continuous Integration		X			X
Daily standup meeting	X	X			X
Iteration Demo			X		X
Iteration Planning	X X X				X
Iteration Retrospective		X			X
On-Site Customer			X	X	
Pair Programming		X			X
Refactoring					X
Short Iterations	X	X	X	X	X
Small Releases	X				X
Simple Design					X
Test-Driven Development	X	X			X
Time-Boxing X					
Unit/JUNIT Testing					X
User Stories		X			X
Velocity/Estimation X		X			X

Another area related to specific practices and structure involved how people resources were allocated to the team on a monthly basis. It was suggested that the iteration cycle and monthly man-loading cycle be synchronized to minimize members from rotating off the team in the middle of an iteration. Thus the following is proposed:

Proposition A4: A successful globally distributed agile team configuration synchronizes the iteration cycle and the monthly man-loading cycle so that members are not rotating off the team in the middle of an iteration

Tables 95 through 97 present summaries of the propositions:

Table 95

Summary of Propositions related to Structure

Propositions related to Structure
Proposition S1: A successful globally distributed agile team configuration emphasizes an understanding of the overall project purpose and goal and provides a vision of the "big picture" in an effort to increase meaningfulness, motivation, and quality
Proposition S2: A successful globally distributed agile team configuration takes into consideration the visibility and criticality of a given project with the understanding that some projects will have a greater or lesser degree of these attributes and thus the meaningfulness to the members will be somewhat dependent upon this level
Proposition S3: A successful globally distributed agile team configuration takes the concepts of accountability, responsibility, trust, motivation, growth, ownership, and challenge into account for the purpose of increasing the likelihood of members to find meaningfulness in the project and to develop a personal interest in it

(table continues)

Table 95 *(continued)*.

Propositions related to Structure

Proposition S4: A successful globally distributed agile team configuration realizes the differences in experience and age as determinants of the level of meaningfulness ascribed to the project by its members

Proposition S5: A successful globally distributed agile team configuration emphasizes the use of agile practices such as short iterations and small releases to facilitate meaningfulness and motivation within the team by alleviating the overwhelmingness that may accompany complex projects and facilitate small victories

Proposition S6: A successful globally distributed agile team configuration allows for a high to moderate level of autonomy dependent upon the experience level of the individual members and the nature of the project itself

Proposition S7: A successful globally distributed agile team configuration allows a substantial level of freedom and flexibility in terms of business and technical decisions as long as those decisions do not change the project scope or alter prioritization thus not diminishing the agility of distributed team members

Proposition S8: A successful globally distributed agile team configuration emphasizes the shared concepts related to autonomy and agility including flexibility, creativity, ownership, accountability, shared responsibility, collaboration, and encouragement

(table continues)

Table 95 *(continued)*.

Propositions related to Structure

Proposition S9: A successful globally distributed agile team configuration facilitates feedback from multiple sources in both verbal and written form, positive or negative, technical and managerial, with a special emphasis on the customer and individual team members to foster team agility

Proposition S10: A successful globally distributed agile team configuration implements appropriate agile practices to glean feedback from the work itself and to facilitate a greater degree of input potentially increasing the agility of the team

Proposition S11: A successful globally distributed agile team configuration allows for the development of core norms of conduct in an evolutionary and natural way that allows for flexibility and trust in the professionalism of its members

Proposition S12: A successful globally distributed agile team configuration clearly defines the norms upfront either explicitly or implicitly and establishes expectation while providing accommodations for lack of physical presence and crossing cultural boundaries

Proposition S13: A successful globally distributed agile team configuration establishes norms not only for personnel related issues but also for technical and performance related issues raised by the use of agile practices and global distribution

(table continues)

Table 95 *(continued)*.

Propositions related to Structure
Proposition S14: A successful globally distributed agile team configuration consists of small teams, of no more than twenty members, in an effort to foster effective communication, coordination, and control and to alleviate cultural differences within a global distributed environment
Proposition S15: A successful globally distributed agile team configuration strives to allocate members 100% to the team and to assign the appropriate skill sets at project initiation with the option of ramping up the team up as necessary
Proposition S16: A successful globally distributed agile team configuration consists of a smaller number of members to better facilitate certain agile practices such as the daily stand-up meeting, iteration planning, iteration demos, iteration retrospectives, and user stories
Proposition S17: A successful globally distributed agile team configuration consists of a balanced mixture of members in terms of personal, managerial, and technical skills to increase the degree of creativity, idea generation, knowledge sharing, innovativeness, and “teamness”

(table continues)

Table 95 *(continued)*.

Propositions related to Structure

Proposition S18: A successful globally distributed agile team configuration includes members with the necessary business, managerial, and technical skills necessary to complete a given project and should take advantage of the large resource pool afforded by a globally distributed environment

Proposition S19: A successful globally distributed agile team configuration takes advantage of the benefits of a balanced team membership and the values and principles associated with agile methodologies

Proposition S20: A successful globally distributed agile team configuration includes a wide range of knowledge and task-related skills in the areas of business, project management, and technical necessary to complete projects and takes advantage of the global nature of the team by drawing from a large pool of specialized resources

Proposition S21: A successful globally distributed agile team configuration acknowledges and accepts the need for compensation and strives to assist members who are not as strong in a particular area until they reach the necessary level of expertise

(table continues)

Table 95 *(continued)*.

Propositions related to Structure
Proposition S22: A successful globally distributed agile team configuration takes advantage of agile practices which promote teamwork and help new or less experienced members to increase their knowledge and task-related skills
Proposition S23: A successful globally distributed agile team configuration seeks members with strong interpersonal skills and encourages the development of these skills in order to take advantage of the many benefits they provide in a globally distributed environment
Proposition S24: A successful globally distributed agile team configuration takes advantage of the agile practices which emphasize daily interaction and utilize these practices to foster stronger interpersonal relationships
Proposition S25: A successful globally distributed agile team configuration acknowledges that projects can still be completed in the absence of strong interpersonal skills but that personal conflicts may cause delays and all differences should be handled in a professional manner

Table 96

Summary of Propositions related to Virtualness

Propositions related to Virtualness
Proposition V1: A successful globally distributed agile team configuration provides at least some degree of overlapping work hours between sites to facilitate the use of agile values, principles, and practices
Proposition V2: A successful globally distributed agile team configuration makes use of multiple communication channels through the use of teleconferencing, instant messaging, desktop sharing, groupware, and email technologies
Proposition V3: A successful globally distributed agile team configuration may cross multiple boundaries in regard to functional, organizational, and cultural without negative affect, but organizational "buy-in" and the use of an agile process by other functional unites are beneficial
Proposition V4: A successful globally distributed agile team configuration consists of a longer team lifecycle to allow for the development of strong personal and professional relationships; creation of stability, familiarity, consistency, and commitment; retention of hard to find skill sets; adjustment to steep learning curve, size, and complexity of projects; and avoidance of the need to ramp the team up

(table continues)

Table 96 *(continued)*.

Propositions related to Virtualness
Proposition V5: A successful globally distributed agile team configuration allows members to play multiple roles within the same team in an effort to take advantages of the multiple benefits it affords
Proposition V6: A successful globally distributed agile team configuration primarily consists of full-time members allocated 100% to one team as opposed to members crossing multiple teams

Table 97

Summary of Propositions related to Agility

Propositions related to Agility
Proposition A1: A successful globally distributed agile team configuration adheres to the “spirit” of the values of the Agile Manifesto either directly or indirectly although individual members may not be familiar with the Agile Manifesto in its official form

(table continues)

Table 97 *(continued)*.

Propositions related to Agility
Proposition A2: A successful globally distributed agile team configuration adheres to the “spirit” of the principles of the Agile Manifesto either directly or indirectly although individual members may not be familiar with the Agile Manifesto in its official form
Proposition A3: A successful globally distributed agile team configuration Implements multiple agile practices through tailoring the practices to its specific needs and uses ICT to effectively overcome the lack of face-to-face interaction through synchronous and asynchronous communication
Proposition A4: A successful globally distributed agile team configuration synchronizes the iteration cycle and the monthly man-loading cycle so that members are not rotating off the team in the middle of an iteration

CHAPTER 5

DISCUSSION AND CONCLUSIONS

Discussion of Research Findings

Propositions were developed in the Chapter 4 in relation to each dimension of the theoretical framework. This chapter discusses the most interesting and relevant findings in regard to those propositions, presents a modified version of the theoretical framework, explains the limitations and contributions of the study, and concludes with recommendations for future research.

Team Agility

This section presents the findings of the study in terms of team agility including values, principles, and practices. Table 98 provides a summary of prior literature related to the agility dimension in distributed agile teams:

Table 98

Summary of Literature Related to Agility Dimension of Distributed Agile Teams

	Fowler (2006)	Hogan (2006)	Nisar & Hameed (2004)	Layman et al. (2006)	Ramesh et al. (2006)	Sepulveda (2003)	Yap (2005)
Agility							
Values							
Principles	X		X				
Practices	X		X		X		

Overall the study supported the findings of existing literature suggesting that agile methodologies can be successfully implemented in a globally distributed environment (e.g., Holstrom et al., 2006; Ramesh et al., 2006; Schummer & Schummer, 2001; Yap, 2005). In terms of the Agile Manifesto it was found that team members for the most part were not familiar with it. However, after showing the Agile Manifesto to the individual team members they acknowledged that the majority of the values and principles were adhered to either directly or indirectly. In other words, the "spirit" of the Agile Manifesto was a part of the team's process and practice whether they recognized them in a formal list. The study suggested that all four of the values could be implemented as well as the majority of the twelve practices. Even principle 6 which states that "the most efficient and effective method of conveying information to and within a development team is face-to-face conversation" (Fowler & Highsmith, 2001) did not pose a significant problem for the teams. These findings were set forth in Propositions A1 and A2.

The study also indicated that none of these teams had adopted a formal agile methodology (e.g., XP or Scrum) in its entirety. Two teams had implemented a limited number of actual practices, two teams had adopted a modified version of Scrum coupled with additional practices commonly considered agile, and one team had adopted a majority, but not all of the practices of XP. The study suggested that multiple practices could be successfully implemented in a globally distributed agile team. This supported the literature which suggests that agile practices can be tailored to be effective in a globally distributed environment (Fitzgerald et al., 2006; Kircher et al, 2001; Schummer & Schummer, 2001). Schummer and Schummer (2001) identified certain agile practices that required colocated teams and those which did not. The authors listed the planning game (iteration planning), pair programming, continuous integration, and on-site customer as those practices requiring a colocated team. This study revealed that even these practices could be implemented in a globally distributed agile team via the appropriate ICT such as teleconferencing, instant messaging, desktop sharing, and shared code repository. This study defined agility as the extent to which the team aligns with the general values and principles of the Agile Manifesto as well as with the practices of a specific method. It could be argued, however, that teams that are adopting certain agile practices to their specific situation are exhibiting greater agility than those teams which are trying to strictly adhere to all practices whether those practices are applicable and or necessary, i.e., “being agile” not simply “doing agile”. This finding was set forth in Proposition A3. Table 99 summarizes the findings in regard to the intersection between agility and structure:

Table 99

Summary of the Team Agility

Sub-dimension	Findings
Values	It is possible for teams to adhere to the four values of the Agile Manifesto, although Value 4 (colocated business people) was sometime difficult in a distributed setting, but could be supported through the use of a customer proxy
Principles	It is possible for teams to adhere to the principles of the Agile Manifesto and Principle 6 (face-to-face communication) was addressed through the use of multiple ICT
Practices	It is possible for teams to implement multiple agile practices of a specific method via ICT, even those typically assumed to require face-to-face interaction

Intersection of Team Agility and Structure

This section presents the findings of the study by discussing the relationship between agility and structure in terms of each of the structural sub-dimensions including task design, core norms, and team composition. Table 100 provides a summary of prior literature related to the structure dimension of distributed agile teams:

Table 100

Summary of Literature Related to Structure Dimension of Distributed Agile Teams

	Fowler (2006)	Hogan (2006)	Nisar & Hameed (2004)	Layman et al. (2006)	Ramesh et al. (2006)	Sepulveda (2003)	Yap (2005)
Structure							
Task Design							
Meaningfulness				X			X
Autonomy	X		X				
Feedback	X		X	X	X	X	X
Core Norms							
Primary							
Secondary							
Composition							
Size		X					X
Mix	X	X	X	X			X
K & M Skills	X						
Interpersonal	X						

Task Design

Meaningfulness. The findings indicated that meaningfulness was considered an important element for a successful globally distributed agile team configuration supporting prior literature on work group and team design (Hackman, 2002, 2005) and distributed agile teams (Yap, 2005). Specifically related to agility the study indicated that certain agile practices such as daily-stand-up meetings, short iterations, small releases, iteration planning, iteration demos, and iteration retrospectives contributed to meaningfulness by alleviating the overwhelming feelings that may accompany large, complex projects. By breaking the project into shorter iterations and small releases the team is able to see each of the smaller pieces as they come along as well as gaining a sense of the overall project purpose and goal, i.e., the “big picture”. In addition, regular meetings such as the daily stand-up, iteration planning sessions, iteration demos, and iteration retrospectives provide the members with a constant stream of interaction with their colleagues and allow for consistent feedback on how the project is progressing. It was suggested that these practices influenced meaningfulness which in turn impacted motivation, personal interests and quality. These findings were put forth in Propositions S1, S3, and S5.

Autonomy. The findings also suggested that autonomy was viewed as an essential element for a successful globally distributed agile team configuration supporting prior literature on work group and team design (Hackman, 2002, 2005) and research related to distributed agile teams (e.g., Fowler 2006; Nisar & Hameed, 2004). The consensus was that this type of team calls for a moderate to high level of autonomy tempered by individual experience level and the nature of the project. In relation to autonomy and agility several shared concepts were identified including flexibility, creativity, ownership, accountability, shared responsibility, collaboration, and encouragement. If members in a globally distributed environment are not granted a high to moderate degree of autonomy the agility of the team can be diminished because the members are waiting for direction from the project management team. As the study suggested in situations where decisions do not alter the scope or priorities of the project members should have a high degree of flexibility. These findings were put forth in Propositions S6, S7, and S8.

Feedback. The findings revealed that feedback was also identified as a crucial aspect of a successful globally distributed agile team configuration supporting prior literature on work group and team design as well as multiple studies conducted among distributed agile teams (e.g., Fowler, 2006; Nisar & Hameed, 2004; Layman et al., 2006; Ramesh et al, 2006; Supulveda, 2003; Yap, 2005). In general feedback from multiple sources was considered important regardless of whether it was positive or negative. The study indicated that agile practices actually contributed to the feedback loop via daily-stand-up meetings, iteration planning sessions, iteration demos, and iteration retrospectives. Because of the emphasis on regular and effective communication in agile methodologies regular feedback can greatly enhance the agility of the team and a lack of agility may diminish agility. As was indicated by one of the participants, globally distributed agile teams are not communicating every few weeks, but more likely every day. These findings were put forth in Propositions S9, and S10.

Core Norms

The findings suggested that the establishment and benefits of core norms to a successful globally distributed agile team configuration were somewhat ambiguous and the difference between primary norms and secondary norms was not emphasized by the teams. This may be due to the fact that core norms are just not as visible in a globally distributed team as indicated by one participant. This is not necessarily in conflict with the literature on work group and team design as much it is the subject of core norms did not appear to be a major concern of the teams. Of the literature reviewed on distributed agile teams core norms were not mentioned specifically.

In general, however, the teams preferred to develop their own norms rather than depend on standard organizational policies and procedures. This seemed to be based upon the impersonal nature of such policies and procedures; whereas, the overall sentiment of the teams was that norms should be developed via personal relationships in a more natural, evolutionary way. In relation to agility it was recommended that norms be established for certain agile practices such as the daily stand-up meeting to ensure it was conducted in an efficient manner and did not go on and on. On the other hand the sentiment for iteration planning, iteration demos, and iteration retrospectives was that time should not be so much of a factor and that each member should have ample opportunity to provide their input and ideas. These finding were put forth in Propositions S11, S12, and S13.

Team Composition

Size. In regard to size the findings of this study supported prior literature from work group and team design (Hackman 2002, 2005) and current research conducted among distributed and agile teams suggesting that globally distributed agile teams be kept small (e.g., Hogan, 2006; Yap, 2005). As Hogan (2006) indicated, smaller teams were found to be more effective. This was also the belief of Hackman (2002) who suggested that teams should even be a little smaller than needed to place positive pressure on the team and to provide a challenge without overwhelming the members.

As indicated by the study, one member recommended that the team be no more than twenty members. This may be a good rule of thumb, but of course each situation is unique and the project management team should carefully consider the number of members to be placed on the team. However, as the team grows it becomes more cumbersome and there is the law of diminishing returns. According to the study, smaller teams promote more effective communication, coordination, and control in a distributed environment which is very important in agile teams (Holmstrom et al., 2006). In addition interpersonal relationships and cultural differences may be alleviated.

Related to the size of the team the findings indicated that a successful configuration also consists of members who are allocated full-time to the team as opposed to varying degrees of allocation. This reduces the possibility that the team will need to ramp up during busy times and thus bring on more members who also may part-time. This was a significant concern of one of the project managers in the study.

In relation to team size and agility the study revealed that the size of the team had an impact on the use of certain agile practices, namely those related to communication and collaboration. As the teams grew larger such practices as the daily stand-up meeting and iteration planning, for example, became less effective due to difficulty in managing the number of members, the time needed to allow all members to provide a status report, and a general loss of focus by the members. As the team grew it was suggested that it be broken into smaller sub-teams each having its own daily stand-up meeting, iteration planning sessions, etc. These finding were put forth in Propositions S14, S15, and S16.

Mix. The study also found that mix was considered an important element in a successful globally distributed agile team configuration. Overall the consensus was that a balanced mixture of members in terms of managerial and technical skills, knowledge and expertise, age, and personality was a contributing factor to a successful configuration. There were no indications that any of the team desired a completely homogenous mixture of members. A balanced mixture was associated with increased creativity, idea generation, innovativeness, and general sense of “teamness”. This finding supported the work group and team design literature and was supported in multiple studies exploring distributed agile teams (e.g., Fowler, 2006; Hogan, 2006; Nisar & Hameed, 2004; Layman et al., 2006; Yap, 2005).

Fowler (2006) recommended having more seasoned members mixed with less experienced members. This team hired quite a few new graduates so it was important that they have people on the team to help and mentor them. Hogan (2006) indicated that a multidisciplinary team consisting of the proper balance of roles contributed greatly to the success of the project; going as far as to say that “mirroring” should be employed where each role should be represented at each location. Mix was also related to size as stated by Yap (2005), “it is very important to have a ‘balanced’ team, that is, each region should be equal in size and technical skill level” (p. 6). This proper mix of members can be benefited by the global nature of the team as needed resources can be located and utilized without the constraint of geographic distance and by emphasizing the values, principles, and practices of agile. These finding were put forth in Propositions S17, S18, and S19.

Related to mix was the allocation of people resources to the team in relation to the agile practice of short iterations. It was suggested that the iteration cycle and monthly man-loading cycle be synchronized to minimize members from rotating off the team in the middle of an iteration. This finding was put forth in Proposition A5.

Knowledge and task-related skills. The findings indicated an overall agreement that knowledge and task-related skills are essential to a successful configuration supporting work group and team design and distributed agile team literature (e.g., Hackman, 2002, 2005; Fowler, 2006). Having the proper business, managerial, and technical expertise was considered very beneficial. However, the overall feeling was that every member did not need to be an expert in all areas of the project. This finding suggested an interrelationship between mixture and knowledge and task-related skill emphasizing the need to choose members with an eye toward the knowledge and experience they can bring to the team. Again, the ability to find the necessary knowledge and skills is helped by the pool of global resources available. A relationship also exists between knowledge and task-related sills and agility. Agile methods emphasize collaboration and teamwork and include practices for compensating for members who might be weaker in certain areas through pair programming for instance. These finding were put forth in Propositions S20, S21, and S22.

Interpersonal skills. Interestingly the findings suggested significant support for strong interpersonal skills among the membership of the team supporting work group and team design and distributed agile team literature (e.g., Hackman, 2002, 2005; Fowler, 2005). Because the teams were distributed and involved members from multiple countries and cultures the ability to communicate verbally and in writing were cited as critical for a successful globally distributed agile team configuration (Fowler, 2005). Additionally, because one of the core values of agile is communication and it utilizes many practices geared toward daily interaction, the possession of adequate interpersonal skills was heightened in these types of teams. However, it was noted that projects can be completed in the absence of members exhibiting strong interpersonal skills but satisfaction with the overall team experience is decreased. These finding were put forth in Proposition S23-S25. Table 101 summarizes the findings in regard to the intersection between agility and structure:

Table 101

Summary of the Intersection of Agility and Structure

Sub-Dimension	Findings
Task Design	Meaningfulness can be enhanced through agile practices
	Autonomy should be high to moderate to emphasize the principle of self-managing teams
	Feedback can be facilitated through multiple agile practices
Core Norms	Should be developed evolutionary and naturally by team itself
	Supporting the principle of self-managing teams
	Should be developed for team interaction and agile practices
Composition	Size should be kept as small as possible to improve use of communication-oriented agile practices
	Mix should be balanced, but diverse to adhere to the principle of technical excellence and motivated members
	Knowledge and task-related skills should be appropriate for tasks supporting the principle of technical excellence
	Strong interpersonal skills are not an absolute, but are preferable, In order to support the principle of self-managing teams

Intersection of Team Agility and Virtualness

This section presents the findings of the study by discussing the relationship between agility and virtualness in terms of each of the virtual sub-dimensions including temporal distribution, boundary spanning, lifecycle, and member roles. Table 102 provides a summary of prior literature related to the virtualness dimension of distributed agile teams:

Table 102

Summary of Literature Related to Virtualness Dimension of Distributed Agile Teams

	Fowler (2006)	Hogan (2006)	Nisar & Hameed (2004)	Layman et al. (2006)	Ramesh et al. (2006)	Sepulveda (2003)	Yap (2005)
Virtualness							
Temporal Distribution							
Time zone	X				X		
Use of ICT	X	X	X	X		X	X
Boundary Spanning	X					X	X
Lifecycle	X						
Member Roles	X	X	X	X			

Temporal Distribution

According to Bell and Kozlowski (2002) temporal distribution refers to the fact that a team is distributed across time and may be separated by only a few hours or by many hours. Espinosa and Pickering (2006) argued the following in terms of the impact of temporal distribution on distributed teams:

> With today's availability of sophisticated collaboration technology and team's increasing experience working globally, geographic distance is no longer a substantial problem for teams as it used to be, whereas time separation remains a great challenge, particularly as more time zones are represented in a team (p. 1).

Although this statement is potentially true of all types of globally distributed teams, the findings of this study indicated that this statement is especially true for globally distributed agile teams.

Temporal distribution was considered a major factor for a successful team configuration and has a significant influence on the agility of the team. Because agile methods promote interaction on a day-to-day basis through the daily stand-up meetings and regular communication through such practices as iteration planning, iteration demos, and iteration retrospectives, having at least some degree of overlapping work hours between distributed locations was considered crucial. On a consistent basis teams indicated that geographic and even cultural differences were not nearly as important as time differences. Although benefits were cited, overall the consensus was that limited or no overlapping hours had a negative affect on the configuration of the team due to the fact that globally distributed agile teams need to have synchronous communication on a very regular basis. In this particular study the temporal distribution of the teams varied fairly significantly. Some teams had potentially 4 to 6 hours of overlapping work hours whereas other teams had no overlap. It was revealed by some members that the fact that at least they were in the same time zone or similar time zone with other locations enabled them to better implement an agile method.

The study also revealed that the use of multiple forms of information and communication technology (ICT) was a great benefit to the team configuration. This finding was supported by multiple studies conducted among distributed agile teams (e.g., Fowler, 2006; Hogan, 2006; Nisar & Hameed, 2004; Layman et al., 2006; Sepulveda, 2003; Yap, 2005). Teleconferencing, instant messaging, desktop sharing, and groupware or collaborative technologies were considered essential. However, such synchronous ICT as teleconferencing and instant messaging were hindered by the lack of overlapping work hours due to time zone differences. As one member indicated, these technologies become useless if there is not someone in their office on the other end to receive them. Thus, pointing to the fact that even the sophisticated technologies referred to above by Espinosa and Pickering (2007) cannot remedy the challenge of significant temporal distribution. Fowler (2006) and Ramesh et al. (2006) also emphasized some degree of time zone synchronization between the sites if possible. This suggests that organizations should not choose distributed locations simply based upon the lowest cost provider, but should take into consideration the time zone differences as well. Do the cost savings of one site account for the lost time in work hours over a location located in a closer time zone.

Bell and Kozlowski (2002) theorized that an “ideal” virtual team would be highly distributed across time. In theory this may make sense when placing virtual teams on a continuum of “ideal” versus “traditional”. However, the findings of this study did not fully support this argument. Although improvements in ICT and the experience of team members working in a global environment have greatly improved a team’s ability to work in such a configuration, the use of an agile method necessitates a greater degree of real-time communication. Thus, if little or no overlapping hours are available due to temporal distribution, the agility of the team is diminished. These finding were put forth in Propositions V1 and V2.

Boundary Spanning

Boundary spanning indicates that the team not only spans time and space boundaries but also functional, organizational, and cultural boundaries. On the continuum proposed by Bell and Kozlowski (2002), the “ideal” virtual team would span these multiple boundaries. In this study teams did indeed span these boundaries to varying degrees. Some spanned all three boundaries while others spanned only one or two. Regardless, the findings indicated that spanning functional, organizational, and/or cultural boundaries did not negatively impact a successful globally distributed agile team configuration. This finding supported the assertion of Bell and Kozlowski and previous research on distributed agile teams (e.g., Fowler, 2006; Sepulveda, 2003; Yap, 2005). This finding was put forth in Proposition V3.

Lifecycle

Bell and Kozlowski contend that the “ideal” virtual team is characterized by a discrete lifecycle, i.e., they are formed and deployed quickly, accomplish their task, disband, and redeploy. Although there was a somewhat mixed response from the teams involved in this study in terms of the appropriate lifecycle for a successful globally distributed agile team configuration, the findings suggested that a longer lifecycle was more beneficial. Although the team desires to have a high degree of agility and flexibility this was not related to the amount of time the team actually stayed together. According to Conboy and Fitzgerald (2004) agility can be defined as “the continual readiness of an entity to rapidly or inherently, proactively or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment” (p. 40). This definition does not refer to agility in the sense of quick formation and deployment and rapid redeployment.

It could be argued that keeping the team together for a longer time frame makes its agility increase because members build strong relationships, develop greater familiarity with each other and the project, create increased stability and consistency, retain difficult to find skill sets, and establish more effective communication, coordination, and collaboration channels which are all desired within agile methodologies. This finding was put forth in Proposition V4.

Member Roles

In terms of member roles Bell and Kozlowski (2002) suggested that members would participate in multiple roles within multiple teams in an “ideal” virtual team. This study explored members playing multiple roles in the same team as well as members playing roles in multiple teams. The findings indicated that members playing roles within the same team did not negatively impact a successful globally distributed agile team configuration as long as the roles were related in some way and in fact suggested that it afforded multiple benefits. This finding was put forth in Proposition V5. The overall consensus for members playing roles in multiple teams was considered a hindrance to a successful globally distributed agile team configuration and was not recommended if at all possible. The desire of the project managers interviewed was to have full-time resources allocated 100% to the team. This finding was put forth in Proposition V6. Table 103 summarizes the findings in regard to the intersection between agility and virtualness:

Table 103

Summary of the Intersection of Agility and Virtualness

Sub-dimension	Findings
Temporal Distribution	Providing for at least some degree of overlap in working hours allows for greater use of agile practices through synchronous and asynchronous communication
	Use of multiple ICT greatly enhances the use of communication-oriented agile practices
Boundary Spanning	It is possible to cross functional, organizational, and cultural Boundaries without negative effect
	It is beneficial if other functional unites are utilizing an agile process and the organization is supportive
Lifecycle	Longer is preferable to emphasize value of individuals and Interactions and self-managing teams
	Shorter does not necessarily equate to higher agility
Member Roles	Preferable to have full-time members, clearly defined roles if Possible, and 100% allocation to the team
	Having team members play multiple roles in multiple teams does not Necessarily equate to higher agility

Limitations of the Study

This section discusses the limitations of the study. First, in regard to the selection of cases it is acknowledged that multiple configurations of globally distributed agile teams exist and that this study only explored five out of those many possibilities.

Second, the cases explored in this study involved only those perceived as successful configurations by the members. Thus, the study did not take into consideration cases where unsuccessful configurations were involved. This raises the question of how teams that perceived themselves as an unsuccessful configuration would impact the results and implications of the study.

Third, there is the potential that data collected based upon the perception of the participants was skewed and/or biased and did not actually represent reality. Due to the exploratory nature of the design the findings of the study were based upon self-reported data.

Fourth, the terminology used in the interview protocol may not have been as familiar to some participants as to others due to the cultural and language differences involved with globally distributed teams.

Fifth, the fact that the interviews were digitally recorded may have influenced the participant’s responses and it is possible that they may not have been as forthcoming if the interviews were not taped.

Sixth, as Yin (2003) suggested, the data analysis procedures associated with the case study method relies “strongly on argumentative interpretation, not numeric tallies”; therefore, the challenge for the researcher is to “know how to develop strong, plausible, and fair arguments that are supported by the data” (p. 135). Considering that this study was conducted by a single investigator, collection, analysis, and interpretation of data were approached from a single point of view subjecting the findings to some degree of potential researcher bias. Subjectivity of interpretation of the data was a part of the data analysis process and thus issues of validity of these interpretations may be considered a limitation.

Seventh, the study did not take into the consideration the specific type of tasks performed by the team beyond the fact that they were involved with software development projects.

Eighth, and related to seven, the complexity of the team's project was also not taken into consideration beyond the understanding that some of the projects were new development and others were maintenance-related. Based upon these two limitations, two questions could be asked: (1) whether or not a successful configuration for one project would be an equally successful configuration in another of a different type, and (2) how does the task complexity of the project affect a successful configuration?

Ninth, the term successful was defined broadly as "a favorable or desired outcome" (Funk & Wagnalls, 1984). In general the participants were asked to comment on what configuration they thought produced the best or optimal results. A limitation of the study may be the fact that data were based upon self-reporting rather than a specific, defined measurement instrument.

Tenth, in operationalizing the term "team agility" a more comprehensive definition might have included more of an emphasis on how and to what extent the values, principles, and practices were actually implemented rather than just assigning a numeric value. It could be argued that agility is based more upon how the team adapts the values, principles, and practices of agile to their specific situation than the fact that they have implemented all of them even when they may not be applicable or necessary.

Finally, a traditional limitation cited for case study research relates to the generalizability of the findings. As such this study does not provide statistical generalization which where “an inference is made about a population (or universe) on the basis of empirical data collected about a sample” (Yin, 2003, p. 32); rather the goal of case study research is analytical generalization which involves generalization to “theoretical propositions and not to populations or universes” (p. 10). However, caution should be taken when generalizing the findings of this study to cases outside of its specific context as indicated by Lee and Baskerville (2003), “hence, a theory generalized from the empirical descriptions in a particular case study has no generalizability beyond the given case” (p. 236). However, Lee and Baskerville concurred with Yin (2003) by stating:

> In summary, the notion of generalizability of empirical descriptions to theory is well-developed. Hence criticisms that case studies and qualitative studies are not generalizable would be incorrectly ruling out the generalizability of empirical descriptions to theory. Furthermore, such criticism could be incorrectly presuming that statistical generalizability is the only form of generalizability (237).

Thus, although a “case study has not generalizability beyond a given case” (Lee & Baskerville, 2003, p. 236), the “researcher may appropriately strive to develop a theory that is generalizable within the case setting” (Lee & Baskerville, p. 241).

Contributions of the Study

Contributions to Research

This section discusses the contributions of the study to research. First, the findings support the previous literature which argues that agile methods can be successfully implemented within globally distributed environments (e.g., Kircher et al., 2001; Ramesh et al., 2006; Schummer & Schummer, 2001; Xiaohu, 2004). Second, it begins to provide answers to the question of how the configuration of globally distributed agile teams differs from the configuration of other types of globally distributed teams. Third, it synthesizes past research and the current findings into a theoretical framework which is grounded in the data that can be utilized by researchers. Finally, and perhaps most importantly, it provides a starting point for theorizing about how agile software development teams can be successfully configured in a globally distributed environments by exploring the dimensions of team structure, virtualness, and agility, as discussed in the theoretical framework. As a result of the study the preliminary theoretical framework set forth in Chapter 2 has been revised slightly to better represent the sub-dimensions of each dimension. Each of the sub-dimensions of team structure, virtualness, and agility has been added to the framework to provide clarity. The decision was also made to remove the “high” and “low” designations from the virtualness dimension since its operationalization based upon the work of Bell and Kozlowski (2002) was found not to be completely reflected by the findings. Similarly, the “high” and “low” designations were removed from the agility dimension since it was determined that its definition should be broader than simply the number of values, principles, and practices employed. The revised theoretical framework is presented in Figure 3:

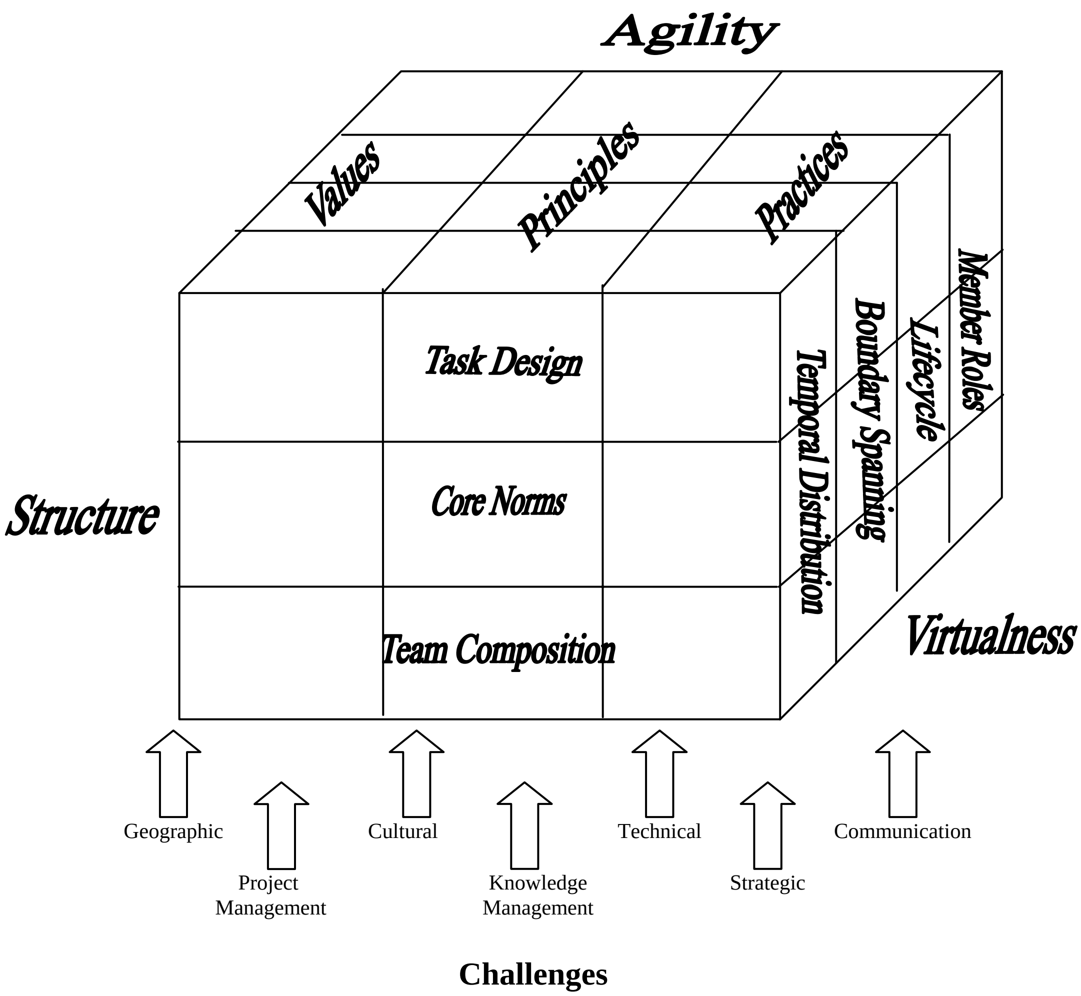

Figure 3. Revised theoretical framework.

Contributions to Practice

Because of the dynamic nature of organizations in the global arena and the reoccurring challenges involved in software development projects, one goal of a successful globally distributed agile team configuration is to be as adaptable and flexible as possible utilizing the strengths of agile methods and global virtual teams. Based upon the revised theoretical framework presented in Figure 3 the overall proposition of this study was that globally distributed agile teams can vary in their possible configurations and that some configurations are more successful than others thus this section discusses the contributions of the study to practice. First, similar to number three of the research contributions, the synthesis of past research and current findings into a theoretical framework can also be beneficial to practitioners. Second, the study can help practitioners to address the challenges related to the configuration of globally distributed agile teams. Finally, and perhaps most important to practitioners, this study presents a set of best practices which organizations can follow when configuring this specific type of team that is informed by both the data and the existing literature. These best practices are presented in Table 104:

Table 104

Best Practices for a Successful Globally Distributed Agile Team Across Five Teams

Best Practices
1. Increase meaningfulness via agile practices like short iterations and small releases
2. Provide a high to moderate degree of autonomy
3. Emphasize regular feedback via agile practices such as daily stand-up meetings, iteration planning, iteration demos, and iteration retrospectives, short iterations, small releases, continuous integration, and frequent builds
4. Establish expectations and roles upfront, but allow core norms to develop naturally among the team itself
5. Keep teams as small as possible or break existing large teams into smaller sub-teams
6. Select a diverse mix of members to facilitate creativity, idea generation, and enthusiasm
7. Pick members with the appropriate knowledge and task-related skills understanding that every member does not need to be an expert in every area of the project
8. Choose members with good interpersonal skills if possible and encourage the development of these skills among all members
9. Select sites with at least some degree of overlapping work hours not simply the low cost location

(table continues)

Table 104 *(continued)*.

Best Practices
10. Utilize multiple ICT with an emphasis on teleconferencing, instant messaging, and desktop sharing
11. Acknowledge that crossing multiple boundaries does not have to negatively impact the team
12. Strive for a longer team lifecycle to allow for the development of stronger working relationships
13. Establish clear roles, but allow members to play multiple roles within the team when necessary to facilitate cross-training
14. Allocate members 100% to the team utilizing full-time resources
15. Synchronize iteration cycle and monthly man-loading cycle so that members are not rotating off the team in the middle of an iteration

Recommendations for Future Research

The following section discusses recommendations for future research. First, this research sets the stage for conducting additional research to gain a deeper understanding of how time zone differences specifically impact globally distributed teams as opposed to other types of globally distributed teams. Second, it is suggested that this research be extended to approach the topic from the perspective of sociotechnical theory for the purpose of identifying social and technical dimensions that will inform organizations in configuring globally distributed teams. Third, in addition to the theoretical lens of Configurational theory, it might be beneficial to incorporate the concepts of Contingency Theory as well considering that it address similar constructs as well as others including efficiency, performance, strategy, technology, task, organizational size, structure, and culture. More specifically the representation of Contingency Theory in MIS research would serve as a potentially interesting theoretical framework as represented in Figure 4:

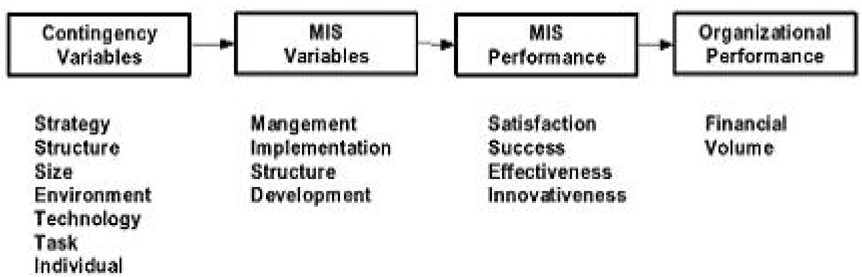

Figure 4. Representation of Contingency Theory in MIS research (n.d.).

Fourth, the study of how the level of agility and level of virtualness impact the team structure itself is another avenue for additional research. This would seek to examine four types of team structure: (1) low agility, low virtualness; (2) high agility, low virtualness; 3) low agility, high virtualness; and (4) high agility, high virtualness. It would be very interesting to see how these different team structures vary from one another based upon the levels of agility and virtualness and how they might impact the use of agile methods in globally distributed environments. The explanation of the configuration of a high agility, high virtualness team would be of interest both in research and practice.

Finally, while conducting this research it was discovered that team processes are another important dimension of team structure which should be considered when configuring globally distributed agile teams. Prasad and Akhilesh (2002) suggested that team processes are an important structural element and consist of several considerations. The first deals with the mechanism for making decisions. This authority to make decisions might be centralized or decentralized. It may be more formal in which decisions lie with the manager or project leader. Or it may less formal in that the team members are given more control over the decisions that must be made. The decision-making process dovetails with task design and the autonomy of the team. The second process addresses the degree of information sharing between the members of the team and their participation in the long-range planning of projects.

The third process deals with the modes of control and communication and coordination. The mode of dealing with these three issues may be very systemized, based upon a strict set of procedures and standards on one extreme or left to the actual team to develop their own mode of achieving the overall goal and dealing with behaviors and interactions of the team members. Or, the mode may be somewhere in between, where guidelines as opposed to stringent rules or little direction.

Finally, the last process involves the degree of commonality in work process and technology infrastructure. This is an important process due to the potential problems caused by strategic issues such as division of work, time zone differences, and technological inconsistencies between distributed sites. In light of this finding a proposed theoretical framework has been developed which is presented in Figure 5.

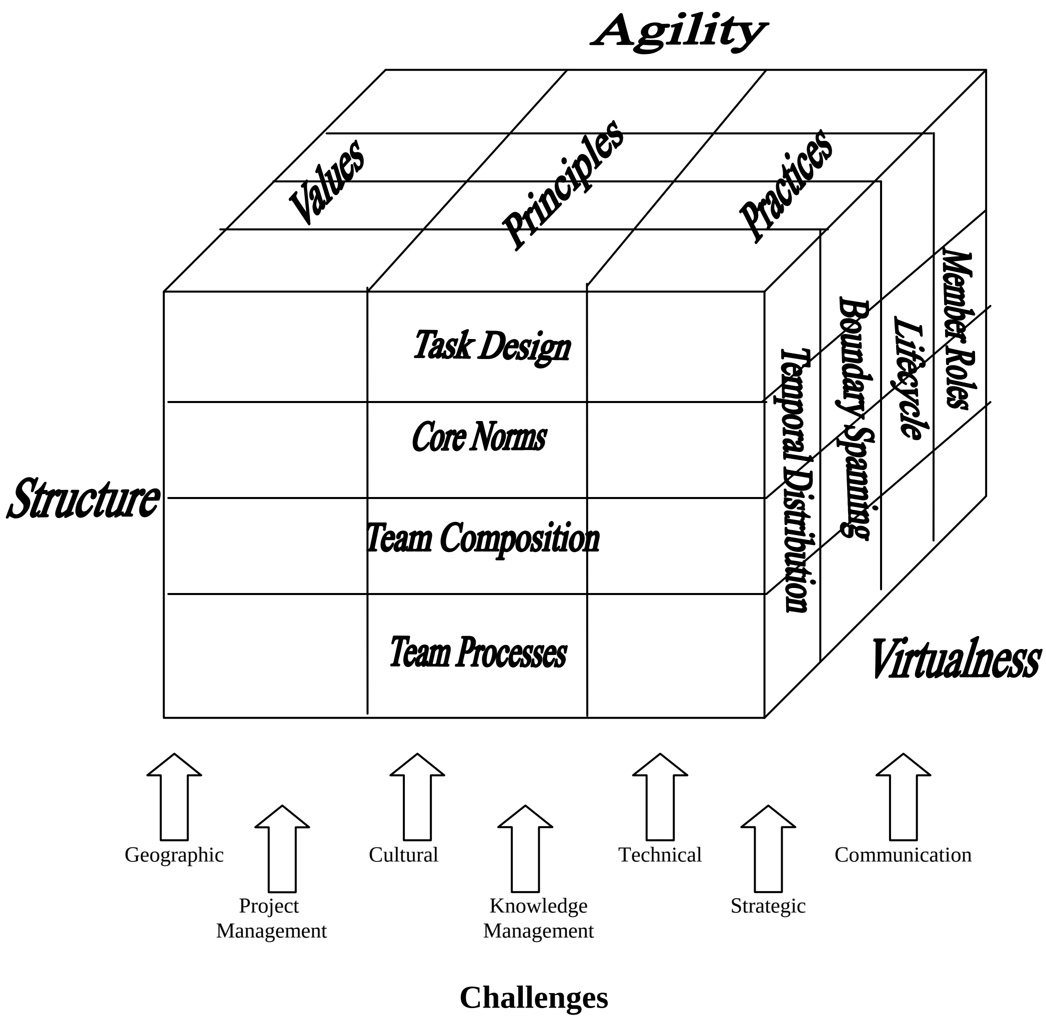

Figure 5. Proposed theoretical framework for future research including team processes.

Conclusion

In conclusion this study presents a framework based on extant literature and grounded in the data for theorizing about how to successfully configure globally distributed agile teams, highlights the potential that globally distributed teams may have to impact the field of software development, provides researchers and practitioners with a better understanding of how the configuration of globally distributed agile teams differ from other types of globally distributed teams, and serves as a building block for further research in this emerging area.

APPENDIX A

INFORMED CONSENT DOCUMENTS

UNT
UNIVERSITY OF
NORTH TEXAS
DISCOVER THE POWER OF IDEAS

RESEARCH AND TECHNOLOGY TRANSFER
Office of Research Services

June 27, 2007

Jason Sharp
Department of Information Technology & Decision Sciences
University of North Texas

RE: Human Subjects Application No. 07-224

Dear Mr. Sharp:

In accordance with 45 CFR Part 46 Section 46.101, your study titled "Globally Distributed Agile Teams: An Exploratory Study of the Dimensions Contributing to Successful Team Configuration" has been determined to qualify for an exemption from further review by the UNT Institutional Review Board (IRB).

Enclosed is the consent document with stamped IRB approval. Please copy and **use this form only** for your subjects.

No changes may be made to your study's procedures or forms without prior written approval from the UNT IRB. Please contact Shelia Bourns, Research Compliance Administrator, ext. 3940, if you wish to make any such changes.

Sincerely,

Scott Simpkins, Ph.D.
Chair
Institutional Review Board

Informed Consent Telephone Script

Hello, my name is Jason Sharp. I am a doctoral student at the University of North Texas. You are being asked to participate in a research study consisting of globally distributed teams utilizing agile software development methods. The purpose of this study is to explore the dimensions contributing to the successful configuration of globally distributed agile teams.

If you choose to be in the study, your participation will consist of an interview that will be audio recorded, describing your experience working in a globally distributed agile team. It is estimated that it will take approximately 60 minutes of your time to answer the questions, and you may be re-contacted for further participation in the future for a follow-up interview of approximately 30 minutes.

To protect your privacy, you will be assigned an identification number. All of the information you provide will be stored only with your identification number, *not* with your name. The audio recordings will be transcribed and analyzed, and will be destroyed after the research is complete. The hardcopy transcription will be stored in a locked cabinet. The audio recordings and the electronic transcription will be kept on a computer that is password protected. Only I will have access to the original audio recordings and transcription. Every effort will be taken to protect the identity of the participants in the study. You will not be identified in any report or publication of this study or its results.

There are no foreseeable risks associated with this study. The benefits, which may reasonably be expected to result from this study, are both for research and practice and will result in better understanding of the successful configuration of globally distributed agile teams. Your decision whether or not to participate in this study will not affect your employment at [Organization Name]. If you agree to participate in this study, please understand that your participation is voluntary. You have the right to withdraw your consent or stop your participation at any time without penalty. You have the right to refuse to answer any particular questions.

If you have any questions regarding this study, please contact Jason Sharp at [redacted] or via email [redacted] or Dr. Sherry Ryan at [redacted] Information Technology and Decision Sciences department or via email: [redacted]

This research study has been reviewed and approved by the Institutional Review Board (IRB) at the University of North Texas. You may contact the UNT IRB at 940-565-3940 with any questions regarding your rights as a research subject.

Do you understand the information which has been presented and agree to freely give your consent to participate in this study and have your responses audio recorded? Do you understand that a copy of this script will be made available to you for the relevant information including contacts?

Thank you for your time. Let us now begin the interview.

APPENDIX B

INTERVIEW PROTOCOL

INTERVIEW PROTOCOL

General Questions – To project lead or manager only

1. How many members are on the team?
2. How were the team members chosen?
3. Where are the individual team members physically located?
4. What is the project(s) the team is working on?
5. How long has the team existed? (Have all members been a part of the team since its inception?)
6. Is this a temporary project team that will be disbanded once the work is finished or is it an ongoing project team that will keep operating indefinitely into the future?
7. What specific agile method(s) is the team using?
8. What specific practices of this agile method(s) are being used?
9. Do the team members recognize the agile methods as such?
10. Are the team members familiar with the values and principles of agile methods such as those provided in the Agile Manifesto?
11. Have the teams had specific training on agile methods?
12. Has the team worked on other projects? If so, were they agile development projects?
13. How would you define a successful team configuration?
14. In your opinion, would you rate this team as successful?
15. Are project documents available for review?

16. Does the organization archive email correspondence between team members? Would it be possible for me to review these archives?

17. Do you have a hierarchy chart for the team listing titles of members? Would it be possible for me to obtain a copy of this chart?

Demographic Questions – To individual team members

1. What is your gender? _____ Female _____ Male

2. What is your age? under 30 31-40 41-50 51-60 60+ (please circle one)

3. What is your nationality?

4. Where are you located?

5. How long have you been employed in your current organization?

_____ Less than 6 months	_____ 5-8 years
_____ 6-12 months	_____ 9-16 years
_____ 1-2 years	_____ 17-24 years
_____ 3-4 years	_____ 25 years or more

6. How long have you been in your present position in this organization?

_____ Less than 6 months	_____ 5-8 years
_____ 6-12 months	_____ 9-16 years
_____ 1-2 years	_____ 17-24 years
_____ 3-4 years	_____ 25 years or more

7. How long have you been a member of the team you described in this interview?

____ Less than 1 month	____ 5-6 months
____ 1 month	____ 6 months-1 year
____ 2 months	____ 1-2 years
____ 3-4 months	____ 3 years or more

8. What is your primary role on the team?

9. What is your title?

10. Is your work on the team just one part of your overall job in this organization or, is working on this team the main part of your job in this organization?

Specific Questions – To individual team members

Team Structure

Task Design

Meaningful/Whole Task

1. Is it important to the members to sense they are working on larger project than just the part that is assigned specifically to the team?
2. Are members more likely to work harder if they sense they are working on a project bigger than their individual part?

Autonomy

1. How do you feel the autonomy of the team influences successful configuration? Do you believe that an individual's feelings about who's in "control" of the project positively or negatively influence the attitudes of members?
2. What level of autonomy do you think is most appropriate in globally distributed agile teams? High? Moderate? Low? Briefly explain why.

Feedback

1. How does the amount of feedback influence the successful configuration of teams?
2. How important is it that the feedback comes directly from the work being done rather than from other people in the organization or vice-versa?

Core Norms

1. How does the establishment of acceptable and unacceptable behavior influence the configuration of the team?
2. How important is it for norms of conduct to be established to address issues such as communication, conflict, and cultural differences for the successful configuration of the team?

Team Composition

Size

1. How does the size of the team influence the successful configuration of the team?
2. In your opinion, does a larger team or small team contribute more to the successful configuration of the team?

Mix

1. How does the mixture of members influence the successful configuration of the team?
2. How important is member diversity in the successful configuration of the team?
3. In your opinion, which team contributes more to a successful configuration: a team with many similar members or with many dissimilar members?
4. In your opinion, how should members be selected for the team?

Knowledge- and Task-Related Skills

1. How does the knowledge- and task-related skills of the members influence the successful configuration of the team?
2. In your opinion, is it advisable/possible for other members to compensate for the lack of skill possessed by specific members of the team?

Interpersonal Skills

1. How do the interpersonal skills of each member influence the successful configuration of the team?
2. In your opinion, how important is it that team members get along interpersonally?

Team Agility

Values and Principles

1. Please describe the overall development philosophy/values that the team ascribes to.
2. Please describe any specific development principles that the team tries to follow.
3. What are the benefits/challenges of adopting a particular development philosophy/set of values and principles encountered in a globally distributed team?
4. Does the team strive to adhere to the overall values espoused in the Agile Manifesto?
5. If so, which values are emphasized more? Which values are emphasized less?
6. Are there simply some values that are nearly impossible to implement in a globally distributed environment?
7. Does the team strive to adhere to the twelve guiding principles outlined by the Agile Alliance?

8. If so, which principles are emphasized more or less?
9. Are certain principles easier to implement in a globally distributed environment? If so, can you specifically identify those?
10. Are certain principles harder to implement in a globally distributed environment? If so, can you specifically identify those?
11. Are certain principles nearly impossible to implement in a globally distributed environment?
12. Do you feel that being able to implement more of the values and principles actually makes the team more agile or less agile in a globally distributed environment?

Practices

1. Please describe any specific development practices that have been implemented for use within the team.
2. Please describe how some development practices are more applicable to globally distributed teams as opposed to colocated teams.
3. What are the benefits/challenges of adopting specific development practices in a globally distributed team?
4. Which specific agile methods are being used?
5. For each of these specific agile methods, how many specific practices are being used?
6. Why are specific practices not being used by the team?
7. Do you feel that the more specific practices that are implemented make the team more or less agile in a globally distributed environment?

Team Virtualness

Temporal Distribution

1. Please describe the benefits/challenges of having team members distributed across space and time.
2. In your experience, have you found that certain types of information and communication technologies are more useful for certain tasks in a globally distributed team?

Boundary Spanning

1. Please describe the types of boundaries that the team members cross (i.e., cultural, functional, and/or organizational).
2. How does the crossing of multiple boundaries affect the completion of projects?

Lifecycle

1. In your experience, do teams with a longer or shorter lifecycle appear to be more successful? Or, is there no difference?
2. Does the team dynamic differ significantly in long-term as opposed to short-term teams?

Member Roles

1. In your experience, when members are required to play multiple roles in the team, does it affect the success of the team either positively or negatively? Or, no affect?
2. In your experience, when members are required to play roles in multiple teams, does this appear to affect the success of the teams?

Successful Configuration

1. In your opinion, what makes a globally distributed agile team configuration successful?
2. In your opinion, would you rate this team as successful?

APPENDIX C

CODING LIST

CODING LIST

Team Structure

Task Design

Meaningfulness

Autonomy

Influence

Level

Feedback

Influence

Source

Team Composition

Size

Mix

Selectio n

Interpersonal skills

Knowledge and task-related skills

Influence

Compensat ion

Core Norms

Intersects Agility

Team Virtualness

Temporal Distribution

Use of ICT

Boundary Spanning

Lifecycle

Member Roles

Roles in One Team

Roles in Multiple Teams

Intersects Agility

Team Agility

Practices

Benefits and Challenges

Influence

Valu es

Principles

Successful Configuration

APPENDIX D

CASE STUDY PROTOCOL

CASE STUDY PROTOCOL

Construct	Potential Data Sources	Questions of the Case
(Background)	Team interviews	Brief description of company
		Brief description of software project
		Team members' geographic location
		Characteristics of team members
		Size of the team
Team Structure Task Design	Team interviews; Documentation of organizational chart, team hierarchy	To what extent is the structure based upon existing team literature?
		How well does a well-structured or poorly structured team affect successful configuration?
		Do elements such as meaningfulness, autonomy, and feedback contribute to successful configuration?

Construct	Potential Data Sources	Questions of the Case
Team Structure		
Core Norms		What is the role of core norms in the successful configuration of teams?
Team Composition		Does team size play a significant part in successful configuration?
		How does the team “mix” affect the successful configuration of the team?
		Are there formal criteria for determining the interpersonal skills of potential team members?
		How do interpersonal skills affect successful Configuration

Construct	Potential Data Sources	Questions of the Case
Team Structure		
Team Composition		Are there formal criteria for determining the task-related knowledge and skills of the members? How does task-related knowledge and skills affect successful configuration?
Team Virtualness		
Temporal Distribution	Team interviews; Documentation of team composition and physical locations, and member roles within the team, what organizational or functional unit does the member belong too?	Does the physical location of the team members impact successful configuration? What types of information and communication technologies are employed?

Construct	Potential Data Sources	Questions of the Case
Boundary Spanning		How do team members feel about having to cross organizational, functional, and cultural boundaries ?
		How does boundary spanning affect the successful configuration of the team?
Lifecycle		How does the team lifecycle affect the success of the configuration?
		Is a discrete lifecycle more likely to contribute to successful configuration as opposed to a continuous lifecycle?
Member Roles		How does having multiple roles in the team or affect successful configuration?

Construct	Potential Data Sources	Questions of the Case
Team Agility		
General values and principles	Team interviews; Review of the team's development process	How does the use of agile methods in globally distributed environments affect the successful configuration of agile teams? Do the general values and principles of agile methods promote or impeded successful configuration? Are some values or principles too difficult to adhere to in global settings? Are certain agile methods more conducive for facilitating successful configurations?

Construct	Potential Data Sources	Questions of the Case
Team Agility		
Specific practices		How few specific practices can be employed and the team still be considered an agile team?
		Are some specific practices just too difficult to implement in globally distributed environments thus affecting successful configuration?
Challenges	Team interviews	What are the overall challenges to globally distributed agile teams?
		How can teams overcome the strategic, cultural, communication, geographic, knowledge management, project management, and technical challenges imposed by global distribution?

Construct	Potential Data Sources	Questions of the Case
Successful Configuration	Team interviews	What makes a globally distributed agile team configuration successful? How would individual members rate the successfulness of the team?

References

Abrahamson, P. (2003). Extreme Programming: First results from a controlled case study. *In Proceedings of the Euromicro*, Antalya, Turkey, September 3-5.

Abrahamson, P., & Koskel, J. (2004). eXtreme programming: A survey of empirical data from a controlled case study. In N. Juristo and F. Shull (eds.), *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering.* Los Alamitos, CA: IEEE Computer Society Press, 73-82.

Abrahamson, P., Warsta, J., Sippon, S. T., & Ronkainen, J. (2003). New directions on agile methods: A comparative analysis. *Proceedings of the 25th International Conference on Software Engineering*, 244-254

Alavi, M., & Carlson, P. (1992). A review of MIS research and disciplinary development. *Journal of Management Information Systems*, 8(4), 45-62.

Alshayeb, M. & Li, W. (2005). An empirical study of system design instability metric and design evolution in an agile software process. *The Journal of Systems and Software*, 74(3), 269-274.

Agerfalk, P. J., & Fitzgerald, B. (2006). Flexible and distributed software processes: Old petunias in new bowls? *Communications of the ACM*, 49(10), 27-34.

Agerfalk, P. J., Fitzgerald, B., Holmstrom, H., Lings, B., Lundell, B., & Conchuir, E. O. (2005). A framework for considering opportunities and threats in distributed software development. *In Proceedings of the International Workshop on Distributed Software Development*, Paris, 29 August, Austrian Computer Society, 47-61.

Andres, H. P. (2002). A comparison of face-to-face and virtual software development teams. *Team Performance Management*, 8(1/2), 39-48.

Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005). Agile project management: Steering from the edges. *Communications of the ACM*, 48(12), 85-89.

Balijepally, V., Mahapatra, R., & Nerur, S. (2006). Assessing personality profiles of software developers in agile development teams. *Communications of the Association for Information Systems*, 18, 55-75.

Barkhi, R., Amiri, A., & James, T. L. (2006). A study of communication and coordination in collaborative software development. *Journal of Global Information Technology*, 9(1), 44-61.

Baskerville, R., Ramesh, B., Levine, L., Pries-Heje, J, & Slaughter, S. (2003). Is Internet-speed software development different? *IEEE Software*, 20(6), 70-77.

Battin, R. D., Crocker, R., Kreidler, J., & Subramanian, K. (2001). Leveraging resources in global software development. *IEEE Software*, 18(2), 70-77.

Beck, K., & Andres, C. (2000). *Extreme programming explained: Embrace change*. Boston, MA: Addison-Wesley.

Bell, B. S., & Kozlowski, S. W. J. (2002). A typology of virtual teams: Implications for effective leadership. *Group & Organization Management*, 27(1) 14-49.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in information systems. *MIS Quarterly*, 11(3), 369-386.

Berg, B. L. (2004). *Qualitative Research Methods for the Social Sciences* (5th ed.). Boston, MA: Pearson.

Boehm, B. (2002). Get ready for agile methods, with care. *IEEE Computer*, 35(1), 64-69.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30-39.

Bonoma, T. V. (1985). Case research in marketing: Opportunities, problems, and a process. *Journal of Marketing Research*, 22(2), 199-208.

Carmel, E. (1999). *Global software teams: Collaborating across borders and time zones*. Upper Saddle River, NJ: Prentice-Hall.

Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *IEEE Software*, 18(2), 22-29.

Cockburn, A. (1998). *Writing effective use cases, the Crystal collection for software professionals*. Addison-Wesley Professional.

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *IEEE Computer*, 34(1), 121-133.

Cohn, M., & Ford, D. (2003). Introducing an agile process to an organization. *IEEE Computer*, 36(6), 74-78.

Conboy, K., & Fitzgerald, B. (2004). Toward a conceptual framework of agile methods: A study of agility in different disciplines. *Proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research*, 37-44.

Coram, M., & Bohner, S. (2005). The impact of agile methods on software project management. *Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, 363-370.

Crampton, C. D. (2001). The mutual knowledge problem and its consequences for dispersed collaboration, *Organization Science*, 12(3), 346-371.

Crampton, 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.

Cusumano, M. A., & Yoffie, D. B. (1999). Software development on Internet time. *IEEE Computer*, 32(10), 60-69

Damian, D., & Moitra, D. (2006). Global software development: How far have we come? *IEEE Software*, 23(5), 17-19.

Desouza, Awazu, & Baloh (2006). Managing knowledge in global software development efforts: Issues and practices. *IEEE Software*, 23(5), 30-37.

Drobka, J., Noftz, D., & Raghu, R. (2004). Piloting XP on four mission-critical projects. *IEEE Software*, 21(6), 70-75.

Dube, L. (2004). Investigating information systems with positivist case study research. *Communications of the Association for Information Systems*, 13, 233-264.

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

Dube, L., & Pare, G. (2001). Global virtual teams. *Communications of the ACM*, 44(12), 71-73.

Dwyer, S., Richard, O. C., Chadwick, K. (2003). Gender diversity in management and firm performance: The influence of growth in orientation and organizational culture. *Journal of Business Research*, 56, 1009-1019.

Ebben, J., & Johnson, A. C. (2005). Efficiency, flexibility, or both? Evidence linking strategy to performance in small firms. *Strategic Management Journal*, 26, 1249-1259.

Ebert, C., & De Neve, P. (2001). Surviving global software development. *IEEE Software*, 18(2), 62-69.

Edwards, H. K., & Sridhar, V. (2003). Analysis of the effectiveness of global virtual teams in software engineering projects. *Proceedings of the 36th Hawaii International Conference on System Sciences*.

Eisenhardt, K. M. (1989). Building theories from case study research. *The Academy of Management Review*, 14(4), 532-550.

Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management*, 16(4), 88-100.

Espinosa, J. A., & Pickering, C. (2006). The effect of time separation on coordination processes and outcomes: A case study. *Proceedings of the 39th Hawaii International Conference on System Sciences*, 1-10.

Evaristo, R. J., Scudder, R., Desouza, K. C., & Sato, O. (2004). A dimensional analysis of geographically distributed project teams: A case study. *Journal of Engineering Technology Management*, 21(3), 175-189.

Eveland, J., & Bikson, T. (1998). Work group structures and computer support: A field experiment. *ACM Transactions on Office Information Systems*, 6(4), 354-379.

Ferrat, T. W., Agarwal, R., Brown, C. V., and Moore, J. E. (2005). IT human resource management configurations and IT turnover: Theoretical synthesis and empirical analys is. *Information Systems Research*, 16(3), 237-255.

Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, 15(2), 200-213.

Galegher, J., & Kraut, R. E. (2004). Computer-mediated communication for intellectual teamwork: An experiment in group writing. *Information Systems Research*, 5(2), 110-138.

Gibson, C. B., & Cohen, S. G. (2003). In the beginning: Introduction and framework. Gibson, C. B., and Cohen, S. G. (eds.), *Virtual teams that work: Creating conditions for virtual team effectiveness*, San Francisco, CA: Jossey-Bass.

Glaser, B., & Strauss, A. (1967). *The discovery of ground theory: Strategies of qualitative research*. London, UK: Widenfeld and Nicholson.

Flor, N. V. (2006). Globally distributed software development and pair programming. *Communications of the ACM*, 49(10), 57-58.

Fowler, M. (2006). Using agile software process with offshore development. Retrieved June 1, 2008, from http://martinfowler.com/articles/agileOffshore.html

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Agilemanifesto.org*. Retrieved April 27, 2006, from http://www.agilemanifesto.org

Fowler, M., & Highsmith, J. (2001). History: The Agile Manifesto. *Agilemanifesto.org*. Retrieved April 27, 2006, from http://www.agilemanifesto.org/history.html

Fruhling, A., & De Vreede, G. (2006). Field experiences with eXtreme programming: Developing an emergency response system. *Journal of Management Information Systems*, 22(4), 39-68.

Germain, E., & Robillard, P. N. (2005). Engineering-based processes and agile methodologies for software development: A comparative case study. *The Journal of Systems and Software*, 75(1-2), 17-27.

Hackman, J. R. (2002). *Leading teams: Setting the stage for great performances*. Boston, MA: Harvard Business School Press.

Hackman, J. R. (1990). *Groups that work and those that don't: Creating conditions for effective teamwork*. San Francisco, CA: Jossey-Bass.

Hackman, J. R. (1987). The design of work teams. J. W. Lorsch, eds., *Handbook of organizational behavior*. Englewood Cliffs: NJ: Prentice-Hall.

Hackman, J. R., & Oldham, G. R. (1980). *Work redesign*. Reading, MA: Addison-Wesley.

Hackman, J. R., & Walton, R. E. (1986). Leading groups in organizations. P. S. Goodman, eds. *Designing effective work groups*. San Francisco, CA: Jossey-Bass.

Herbsleb, J. D., & 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. D., & Moitra, D. (2001). Global software development. *IEEE Software*, 18(2), 16-20. Hunt, A., & Thomas, D. (2000). *The pragmatic programmer*. Addison Wesley.

Highsmith, J. (1999). *Adaptive Software Development: A collaborative approach to managing complex systems*. New York, NY: Dorset House Publishing.

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *IEEE Computer*, 34(9), 120-122.

Hilkka, M. –R., Tuure, T., & Matti, R. (2005). Is extreme programming just old wine in new bottles: A comparison of two cases. *Journal of Database Management*, 16(4), 41-61.

Hoffer, J. A., George, J. F., & Valacich, J. S. (2005). *Modern Systems Analysis* (4th ed.). Upper Saddle River, NJ: Prentice Hall.

Hogan, B. (2006). Lessons learned from an eXtremely distributed project. *Proceedings of the AGILE 2006 Conference*, 1- 6.

Hollingshead, A., McGrath, J., & O'Connor, K. (1993). Group task performance and communication technology: A longitudinal study of computer-mediated versus face-to-face groups. *Small Group Research*, 24(3), 307-33.

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

Holstrom, H., Conchuir, E. O., Agerfalk, P. J., & Fitzgerald, B. (2006). Global software development challenges: A case study on temporal, geographical, and socio-cultural distance, *In Proceedings of the IEEE International Conference on Global Software Engineering*.

Hunt, A., & Thomas, D. (2003). The trip-packing dilemma. *IEEE Software*, 20(3), 106-107.

Huo, M., Verner, J., Zhu, L., & Ali Babar, M. (2004). Software quality and agile methods. *Proceedings of the 28th Annual Computer Software and Applications Conference*, 1, 520-525.

Jarvenpaa, S. L., Knoll, K., & Leidner, D. E. (1998). Is anybody out there? Antecedents of trust in global virtual teams. *Journal of Management Information Systems*, 14(4), 29-64.

Jarvenpaa, S. L., & Leidner, D. E. (1999). Communication and trust in global virtual teams. *Organization Science*, 10(6), 791-815.

Johannsen, C., Dittrich, Y., & Juustila, A. (1999). Software engineering across boundaries: Student project in distributed collaboration. *IEEE Transactions on Professional Communication*, 42(4), 286-296.

Jones, M. C., Cline, M., & Ryan, S. (2006). Exploring knowledge sharing in ERP implementation: An organizational culture framework. *Decision Support Systems*, 41(2), 411-434.

Kaiser, P., Tullar, W., & McKowen, D. (2000). Student team projects by Internet. *Business Communication Quarterly*, 63(4), 75-82.

Kayworth, T. R., & Leidner, D. E. (2001-2002). Leadership effectiveness in Global Virtual Teams. *Journal of Management Information Systems*, 18(3), 7-40.

Kayworth, T., & Leidner, D. (2001). The global virtual manager: A prescription for success. *European Management Journal*, 18(2), 183-194.

Ketchen, D. J., Thomas, J. B., and Snow, C. C. (1993). Organizational configurations and performance: A comparison of theoretical approaches. *Academy of Management Journal*, 36(6), 1278-1313.

Kircher, M., Prashant, J., Corsaro, A., & Levine, D. (2001). Distributed eXtreme programming. Marchesi, M., Succi, G., Wells, D., Williams, L., Wells, J. D., eds. *Extreme Programming Perspectives*. Addison-Wesley, Reading, MA. Retrieved September 18, 2006, from http://www.kircher-schwanninger.de/ michael/publications/xp2001.pdf

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), 67-94.

Komi-Sirvio, S., & Tihinen, M. (2005). Lessons learned by participants of distributed software development. *Knowledge and Process Management*, 12(2), 108-122.

Kotlarsky, J., & Oshri, I. (2005). Social ties, knowledge sharing and successful collaboration in globally distributed development projects. *European Journal of Information Systems*, 14(1), 37-48.

Landau, S. I. (Ed.). (1977). *Funk & Wagnall's Standard Desk Dictionary* (Rev. ed.). New York : Funk & Wagnalls.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical. *Biometrics*, 33(1), 159-174.

Layman, L., Williams, L., Damian, D., & Bures, H. (2006). Essential communication practices for Extreme Programming in a global software development team. *Information and Software Technology*, 48(9), 781-794.

Lee, A. (1989). A scientific methodology for MIS case studies. *MIS Quarterly*, 13(1), 33-50.

Lee, A. S., & Baskerville, R. L. (2003). Generalizing generalizability in information systems research. *Information Systems Research*, 14(3), 221-243.

Lim, L. K. S., Acito, F., Rusetski, A. (2006). Development of archetypes of international marketing strategy. *Journal of International Business Studies*, 37, 499-524.

Lindstrom, L., & Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information Systems Management*, 21(3), 41-52.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., & Kahkonen, T. (2004). Agile software development in large organizations. *IEEE Computer*, 37(12), 26-34.

Lipnack, J., & Stamps, (2000). *Virtual teams: People working across boundaries with technology, 2nd ed.*, New York, NY: John Wiley & Sons.

Lipnack, J., & Stamps, J. (1997). *Virtual teams: Reaching across space, time, and organizations with technology*. New York, NY: John Wiley & Sons.

Lurey, J. S., & Raisinghani, M. S. (2001). An empirical study of best practices in virtual teams. *Information & Management*, 38(8), 523-544.

Lycett, M., Macredie, R. D., Patel, C., & Paul, R. J. (2003). Migrating agile methods to standardized development practice. *IEEE Computer*, 36(6), 79-85.

Malhotra, A., Majchrzak, A., Carman, R., & Lott, V. (2001). Radical innovation without collocation: A case study at Boeing-Rocketdyne. *MIS Quarterly*, 25(2), 229-249.

Markus, M. L. (1983). Power, politics, and MIS implementation. *Communications of the ACM*, 26(6), 430-444.

Maznevski, M. L., & Chudoba, K. M. (2000). Bridging space over time: Global virtual team dynamics and effectiveness. *Organization Science*, 11(5), 473-492.

McDonough, E. F., Kahn, K. B., & Barczak, G. (2001). An investigation of the use of global, virtual, and colocated new product development teams. *The Journal of Product Innovation Management*, 18(2), 110-120.

Meso, P., & Jain, R. (2006). Agile software development: Adaptive system principles and best practices. *Information Systems Management*, 23(3), 19-30.

Meyer, A. D., Tsui, A. S., & Hinings, C. R. (1993). Configurational approaches to organizational analysis. *Academy of Management Journal*, 36(6), 1175-1195.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook.* Beverly Hills, CA: Sage Publications.

Miles, R. E., & Snow, C. C. (1978). *Organizational strategy, structure, and process*. New York: McGraw-Hill.

Miller, D. (1987). The genesis of configuration. *The Academy of Management Review*, 12(4), 686-701.

Mintzberg, H. T. (1983). *Structure in fives: Designing effective organizations*. Englewood Cliffs, NJ: Prentice-Hall.

Mintzberg, H. T. (1979). *The structuring of organizations*. Englewood Cliffs, NJ: Prentice-Hall.

Mish, F. C. (Ed.). (2003). *Merriam-Webster's Collegiate Dictionary* (11^{th} ed.). Springfield, MA: Merriam-Webster, Incorporated.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 73-78.

Nisar, M. F., & Hameed, T. (2004). Agile methods handling offshore software development issues. Proceedings of IMMIC, 8th International Multitopic Conference. December 417-422.

O'Leary, M. B., & Cummings, J. N. (2007). The spatial, temporal, and configurational Characteristics of geographic dispersion in teams. *Management Information Systems Quarterly*, 31(3), 433-452.

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research*, 2(1), 1-28.

Olson, G. M., & Olson, J. S. (2000). Distance matters. *Human - Computer Interaction*, 15(2/3), 139-178.

Orum, A. M., Feagin, J. R., & Sjoberg, G. (1991). Introduction: The nature of case study, in Orum, A. M., Feagin, J. R., & Sjoberg, G. (eds), *A case for the case study*, The University of North Carolina Press, Chapel Hill, NC.

Ostroff, C., & Schmitt, N. (1993). Configurations of organizational effectiveness and efficiency. *Academy of Management Journal*, 36(6), 1345-1361.

Palmer, S., & Felsing, J. (2002). *A practical guide to Feature-Driven Development*. Upper Saddle River, NJ: Prentice-Hall.

Pare, G. (2004). Investigating information systems with positivist case study research. *Communications of the Association for Information Systems*, 13, 233-264.

Pare, G. (2002a). Implementing clinical information systems: A multiple-case study within a US hospital. *Health Services Management Research*, 15(2), 71-92.

Pare, G. (2002b). Enhancing rigor of qualitative research: Application of a case methodology to build theories of IT implementation. *The Qualitative Report*, 7(4), Retrieved February 21, 2006, from http://www.nova.edu/ssss/QR/QR7-4/ pare.html

Pare, G., & Elam, J. J. (1997). Using case study research to build theories of IT implementation" in A. S. Lee, J. Liebenau, and J. I. Degross (eds.) *Information Systems and Qualitative Research*, London: UK, Chapman & Hall, 542-568.

Patton, M. Q. (2002). *Qualitative Research and Evaluation methods* (3rd ed.). Thousand Oaks, CA: Sage Publications.

Patton, M. Q. (1987). *How to use qualitative methods in evaluation*. Newbury Park: Sage Publications.

Powell, A., Piccoli, G., & Ives, B. (2004). Virtual teams: A review of current literature and directions for future research. *The DATA BASE for Advances in Information Systems*, 35(1), 6-36.

Qureshi, S., & Vogel, D. (2001). Adaptiveness in virtual teams: Organisational challenges and research directions. *Group Decision and Negotiation*, 10(1), 27-46.

Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006). Can distributed software development be agile? *Communications of the ACM*, 49(10), 41-46.

Reifer, D. J. (2002). How good are agile methods? *IEEE Software*, 19(3), 16-18.

Representation of Contingency Theory in MIS Research (n.d.). Retrieved June 30, 2008, from http://www.fsc.yorku.ca/york/istheory/wiki/index.php /Image:Cont1.JPG

Robey, D., Khoo, H., & Powers, C. (2000). Situated learning in cross-functional virtual teams. *IEEE Transactions on Professional Communications*, 43(1), 51-66.

Sarker, S., & Sahay, S. (2004). Implications of space and time for distributed work: An interpretive study of US-Norwegian systems development teams. *European Journal of Information Systems*, 13(1), 3-20.

Sarker, S., Lau, F., & Sahay, S. (2001). Using an adapted grounded theory approach for inductive theory building about virtual team development. *DATA BASE for Advances in Information Systems*, 32(1), 38-56.

Schummer, T., & Schummer, J. (2001). Support for distributed teams in eXtreme programming. Succi, G., Marchesi, M., eds. *Extreme Programming Examined*, Reading, MA: Addison-Wesley. Retrieved September 18, 2006, from http://www.ipsi.fraunhofer.de/~publications/concert/2001/xp00.pdf

Schwaber, K., & Beedle, M. (2002). *Agile software development with scrum*. Upper Saddle River, NJ: Prentice-Hall.

Sepulveda, C. (2003). Agile development and remote teams: Learning to love the phone. *Proceedings of the Agile Development Conference*, 1-6.

Sillitti, A., Ceschi, M., Russo, B., & Succi, G. (2005). Managing uncertainty in requirements: A survey in documentation-driven and agile companies. *Proceedings of the 11th IEEE International Software Metrics Symposium*, 17-26.

Smith, H. L., Shortell, S. M., & Saxberg, B. O. (1979). An empirical test of the configurational theory of organizations. *Human Relations*, 32(8), 667-688.

Solomon, C. M. (1995). Global teams: The ultimate collaboration. *Personnel Journal*, 74(9), 49-58.

Stapleton, J. (1997). *Dynamic systems development Method, version 3*. Addison Wesley.

Straus, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*, 2nd ed. Sage Publications, Thousand Oaks, CA.

Suchan, J., & Hayzak, G. (2001). The communication characteristics of virtual teams: A case study. *IEEE Transactions on Professional Communication*, 44(3) 174-186.

Townsend, A. M., DeMarie, S. M., & Hendrickson, A. R. (1998). Virtual teams: Technology and the workplace of the future. *Academy of Management Executive*, 12(3), 17-29.

Turk, D., France, R., & Rumpe, B. (2005). Assumptions underlying agile software-development processes. *Journal of Database Management*, 16(4), 62-87.

Van Ryssen, S., & Godar, S. (2000). Going international without going international: Multinational virtual teams. *Journal of International Management*, 6, 49-60.

Wageman, R., Hackman, J. R., Lehman, E. (2005). Team diagnostic survey: Development of an instrument. *The Journal of Applied Behavioral Science*, 41(4), 373-398.

Xiaohu, Y., Bin, X., Zhijun, H., Maddineni, S. R. (2004). Extreme programming in global software development. *Canadian Conference on Electrical and Computer Engineering*, 4, 1845-1848.

Yap, M. (2005). Follow the sun: Distributed Extreme Programming development. *Proceedings of the Agile Development Conference*, 1-7.

Yin, R. K. (2003). *Case study research: Design and methods* (3rd ed.). Thousands Oaks, CA: Sage Publications.