IOWA STATE UNIVERSITY
Digital Repository

2012

The Impact of Process on Virtual Teams: A Comparative Analysis of Waterfall and Agile Software Development Teams

Sondra Ashmore
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Business Administration, Management, and Operations Commons, Engineering Commons, and the Organizational Behavior and Theory Commons

www.manaraa.com

The impact of process on virtual teams: A comparative analysis of waterfall and agile

software development teams

by

Sondra Ashmore

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Human Computer Interaction

Program of Study Committee:
Anthony Townsend (Major Professor)
Brian Mennecke
Frederick Lorenz
Samuel DeMarie
Stephen Gilbert

Iowa State University

Ames, Iowa

2012

Copyright © Sondra Ashmore, 2012. All rights reserved.

TABLE OF CONTENTS

iii

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

ABSTRACT

Studies have shown that task type, social context, and time mediates virtual teams (Martins, Gilson, & Maynard, 2004; Townsend, Hendrickson, & DeMarie, 2002), yet no studies have been conducted comparing the process that the virtual teams are using to complete a task. Two global virtual teams from the same corporation who are using two different software development methodologies, waterfall and agile, were compared to understand the impact that process has on virtual teams. Interviews were conducted with both teams and their responses were coded using the constructs in Adaptive Structuration Theory. The results show that the software development process used by a virtual team does impact the team's culture, orientation toward change, and ultimately the quality of the product they are developing. Careful consideration should be made by software development organizations when deciding which development process they should deploy, given the important implications for virtual team dynamics and product outcomes.

CHAPTER 1. INTRODUCTION

1a. Introduction

The study of virtual teams has been an important area of research that has spanned many different disciplines, with the most prominent being Management, Computer-Mediated Communication, and Computer-Supported Cooperative Work. Each group has used their own unique lens to study the phenomena of virtual teams. These perspectives have added to our current understanding of how this model of teaming has evolved, the challenges these teams face, and the factors that lead to the success of a virtual team.

While significant progress has been made in virtual teaming research, there are still opportunities to fill in gaps that exist today. These are important gaps to fill, given the exponential growth of virtual teams in organizations worldwide. Martins, Gilson, and Maynard (2004) did an extensive analysis on what has been learned about virtual teams and the opportunities that remain to enhance our knowledge of these teams. They cite deficiencies in the following areas: team inputs, team processes, team outcomes, methodological and theoretical issues, planning processes, organizational context, action processes (primarily compared to the same actions in face-to-face interactions), and interpersonal processes such as affect management and group emotion. The present study does not seek to answer all of the open questions that remain about virtual teams, but rather will focus on the team processes and the methodological and theoretical aspects of virtual team research. Powell, Piccoli, and Ives (2004) found similar gaps in virtual team research, stating that ". . . the nature of the team project and its interaction with other team design

variables has not been addressed in previous research" (p. 14). The methodological and theoretical issues that are highlighted include the over-emphasis on empirical research that compares virtual teams to face-to-face teams, the abundance of studies that use small groups of college students in laboratory settings, and the direct effects of "virtualness" on a team which includes both mediating and moderating factors. Powell et al. (2004) argue that overcoming these methodological limitations will yield a greater understanding of organizational power, culture, and the structure that affects the functioning of virtual teams.

Using DeSanctis and Poole's (1994) Adaptive Structuration Theory (AST) as a theoretical model, this study takes team process one step further to not only look at team process as the planning that is involved with achieving the team goal, but with the team structure that prescribes how the planning for the project must be done.

This study investigated an organization that has two large virtual teams developing enterprise software who have traditionally used the waterfall methodology. The first team is continuing to use the waterfall approach that organizes the process in a sequential order with each major phase completing before the next begins. The second team has moved toward a contemporary approach that is based on the idea of iterative development phases called agile development. While some similarities can be found between the two approaches, such as their focus on quality, very different techniques and values are needed when executing waterfall and agile development.

This study investigates the structure that a virtual team uses and the mediating and moderating effects that this structure has on team dynamics. The use of AST as a structural model provides an important framework for understanding the structural differences between the two virtual teams and how these operational structures impact their team dynamics. AST has been widely used in virtual team research (Raghuram, Tuertscher, & Garud, 2010) as well as software development studies that involve virtual teams (Cao, Mohan, Xu, &

Ramesh, 2009; Majchrzak, Rice, Malhotra, King, & Ba, 2000). AST includes "instrumentation choices" which take into account the computer-mediated communication tools that influence the interaction.

The second important contribution of this study is to address some of the prevalent methodological issues that have been pervasive in virtual team research. This study uses organizational workers from a large information technology company who are working on long-term virtual team projects. This study does not compare the virtual teams to face-to-face teams, or teams of student subjects, but instead compares two similar professional virtual teams that differ only in the team process structure they are using. The use of such teams provides a better understanding of the organizational culture and the impact of structure on the virtual team dynamics. This also allows for more generalized results that are beneficial to a typical virtual team that would be found in an organizational setting.

Finally, this study is the first to directly compare the team dynamics of waterfall and agile software development teams.

1b. Literature Review

This chapter explores three major areas of research--virtual teams, product development processes, and AST. The chapter begins by defining virtual teams, and then reviews the emergence of virtual teams and the technological advancements that made virtual teaming possible. Currently accepted success factors for virtual teams are discussed as well as the challenges that virtual teaming presents.

The chapter continues by reviewing the literature on product development processes. The benefits of using a product development process are discussed as well as the differences between the waterfall approach and the iterative agile approach. Best practices and key obstacles are summarized for each approach.

Finally, AST is introduced as a theoretical framework for understanding how process impacts virtual teams. The chapter concludes with a discussion of the propositions that are generated based on the literature review.

1c. Methodology

The methodology used to carry out this study is described in this chapter. The participants in this study work for a large information technology corporation. Two separate virtual teams were studied--one using the waterfall approach and one using the agile approach. A description of the case study approach and the procedures to collect and analyze the data are outlined.

1d. Results

The results chapter shows the data collected from the case study interviews. The first section is the demographic information collected about the participants from the study. Second, an analysis of the interviews is presented, indicating if the interview responses support, refute, or fail to support or refute the propositions.

1e. Discussion

Finally, the discussion chapter gives a detailed analysis of the findings with an emphasis on support for findings in previous research as well as new contributions. Unexpected findings and those worthy of continued investigation are highlighted.

Limitations to the current study and opportunities for future study are discussed. Real world implications for the findings in this study are offered for teams that are seeking to understand how the process they select will impact the dynamics of their virtual teams. Key findings are outlined.

CHAPTER 2. LITERATURE REVIEW

2a. Research Context

As virtual teams have evolved, they have taken on a variety of names and focus areas. Raghuram, Tuertscher, and Garud (2010) have identified six distinct terms used to refer to a virtual team: 'telework', 'telecommute', 'virtual work/team', 'distance work/team', 'distributed work/team', and 'computer-mediated work/team'. The term "virtual teams" was the most highly cited term and connects most heavily to the key clusters in this body of research. Johnson, Heimann, and O'Neill (2001) cited virtual teams as the high-tech office term in the family of new virtual buzzwords which also includes virtual reality, virtual space, and virtual organizations. This study will use the term virtual team to encompass all variations of this type of work.

The field of virtual teaming has changed dramatically over the last twenty years. Raghuram, Tuertscher, and Garud's (2010) research suggests that the field has shifted from an urban planning and transportation focus to a focus on the dynamics and outcomes of virtual teams. The growth in research has been consistent with the growth of virtual teams in organizations. Gartner (2006) estimated that 75% of knowledge-based global projects will be performed by virtual teams by 2015. Powell, Piccoli, and Ives (2004) did a literature review of virtual teaming literature from the mid-1990s to 2004 in terms of the inputs, socio-emotional processes, task processes, and outputs. The key focus areas noted on virtual team research during this time period are represented in Figure 1.
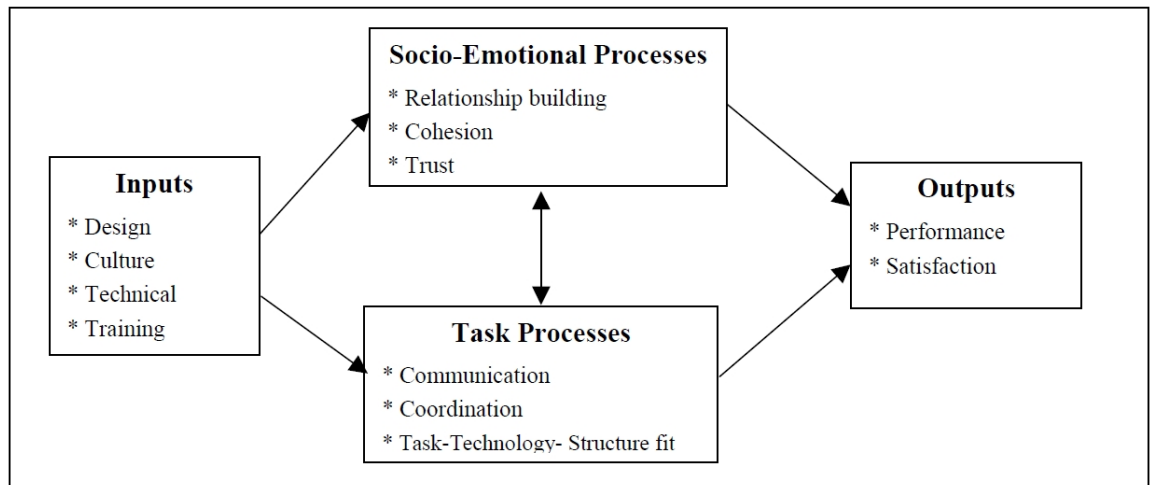
Figure 1: Powell, Piccoli, and Ives (2004) findings on early virtual team research

Powell, Piccoli, and Ives' (2004) findings show there was significant focus on the relational side of teams as well as matching the task with the technology for optimal team performance. Raghuram, Tuertscher, and Garud (2010) results found the key clusters of research disciplines that relate to virtual teams are dynamics of behavior and attitudes, interpersonal relationships, and outcomes.

i) Defining Virtual Teams

Early definitions of virtual teams focused on how physically-distributed virtual teams differ from face-to-face teams through their use of technology (Hertel, Geister, & Konradt, 2005; Martins, et al., 2004). For example, Townsend, DeMarie, and Hendrickson's (1998) early definition of virtual teams stated that they are "groups of geographically and/or organizationally dispersed co-workers that are assembled using a combination of telecommunications and information technologies to accomplish an organizational task" (p. 18). Similarly, Lipnack and Stamps (1997) differentiate virtual teams from face-to-face teams by defining virtual teams as those that "work across space, time, and organizational

boundaries with links strengthened by webs of communication technologies" (p. 7). Other definitions focused on the physical distance of the virtual team members by defining "global virtual teams" (e.g. Jarvenpaa, Knoll, & Leidner, 1998; Jarvenpaa & Leidner, 1999; Kayworth & Leidner, 2002; Maznevski & Chudoba, 2000) and expanding traditional definitions by including a focus on making or implementing decisions central to the organization's global strategy and the fact that team members reside in different countries.

Contemporary definitions of virtual teams weave together elements from definitions presented in seminal works on virtual teams. Martins, Gilson, and Maynard (2004) state that the focus has shifted from differentiating virtual from face-to-face teams and the technology that enables them to the degree of "virtualness" that a team demonstrates. They define virtual teams as "teams whose members use technology to varying degrees in working across locational, temporal, and relational boundaries to accomplish an interdependent task" (p. 808). Powell, Piccoli, and Ives (2004) offer similar groups of geographically, organizationally and/or time dispersed workers brought together by information and telecommunication technologies to accomplish one or more organizational tasks (p. 7). These definitions continue to focus on the geographic and temporal dispersion of a team that has come together through technology to accomplish a task.

Global virtual teams include team members from more than one country (Jarvenpaa & Leidner, 1999; Maznevski & Chudoba, 2000). These teams leverage geographically dispersed expertise (Townsend, et al., 1998) and perspectives, while also allowing for around-the-clock progress on tasks due to time zone differences. Some organizations have made the strategic decision to move to global virtual teams due to lower cost resources (Ellram, Tate, & Billington, 2008) and a local presence in various geographic areas around

the globe (Snow, Snell, Davison, & Hambrick, 1996); however, research has shown that global virtual teams show lower performance than co-located teams (McDonough, Kahnb, & Barczaka, 2001).

ii) Technology Enabling Virtual Teams

The evolution of virtual teams has been strongly guided by technology advancements that enabled remote work environments (Townsend, et al., 1998). One of the earliest technologies that contributed to this movement was the telephone. Co-workers were no longer required to be geographically located at their work place to discuss business issues and participate in meetings. In the late 1990s, Townsend, DeMarie, and Hendrickson stated that video conferencing, collaborative software, and the Internet/Intranet opened new possibilities for geographically dispersed team members much in the same way that personal computers changed day-to-day work patterns of the average workers in the 1980s and 1990s (Townsend, et al., 1998; Townsend, et al., 2002). Recently, cellular phone and tablet technology have further advanced the feasibility of virtual work (Suchan & Hayzak, 2001). According to Maznevski and Chudoba (2000) the most effective communication technology is shaped by the dimensions of the team's task and context (p. 474).

Telecommunications devices are among the most important tools for virtual workers. From the traditional land line phone to conference calls to cell phones, all play a central role in the success of a virtual worker. Virtual team research has found that the richer the media used in the interaction, the more cohesive the feeling of team (Fiol & O'Connor, 2005), the more social presence the team member feels (Baker, 2002), and more trust develops between team members (Warkentin & Beranek, 1999). Given that telecommunications devices rate relatively high on the media richness scale, require little to no training, and are cost effective, it is no surprise that they have remained one of the most popular forms of communication between team members. The advances in cellular phone technology have made it possible to

reach someone by phone no matter where they are in the world. Cellular phone technology, such as that found in the iPhone, even allow for real-time video communications which serves to enhance the media richness.

Collaborative software covers a broad range of options for virtual teaming and is studied extensively in the field of Computer Supported Collaborative Work (CSCW). CSCW is a relatively new field that emerged in the mid-1980s as an area that was primarily interested in the interdisciplinary nature of small group collaboration using technology (primarily computers) (Galegher & Kraut, 1994). While CSCW often is cast as a special interest area in the fields that have helped to evolve its research base such as social psychology and organizational theory (Galegher & Kraut, 1994; Poltrock & Grudin, 1998), it has subsequently established itself as a field worthy of standing on its own (Schmidt & Bannon, 1992). There has been a significant shift toward collaboration using web and mobile technologies since the late 1990s (Bellotti & Bly, 1996; DiMicco et al., 2008; Wagstrom et al., 2011; Westerlund, Normark, & Holmquist, 2011); however, the basic foundation of technology-based collaboration has remained intact. In the context of virtual teams, collaborative software traditionally includes collaboration portals, team rooms, wikis, instant messaging or chat programs, and video conferencing using programs such as Skype. Studies involving virtual teams using collaborative software have found that teams tend to use software that has "media stickiness". Sticky software is difficult to transition away from once it has been implemented (Huysman et al., 2003). Suchan and Hayzak (2001) found that the tool that fit the company's strategy tended to be the one most widely accepted by the team.

The Intranet/Internet has also been a pivotal technology that has significantly expanded the opportunities for virtual teams. The Internet has fundamentally changed the way all team members communicate--offering e-mail, instant messaging, web-based collaboration tools, and video conferencing to name a few. Internet technologies for virtual

teams have engendered varying team preferences and results. Some have argued that e-mail is the most predominantly used and effective medium for virtual teams for information exchange, while insisting that video conferencing is the most valuable for helping build social structures (Townsend, et al., 1998). Others have argued that it is less about the technology choice, and more about how team members choose to communicate over these mediums (Ashmore & Townsend, 2011; Jarvenpaa & Leidner, 1999; Montoya-Weiss, Massey, & Song, 2001).

iii) Best Practices and Challenges

Extensive research has been conducted on factors that contribute to the success of virtual teams. Similarly, the challenges introduced by the use of virtual teams have received a significant amount of attention from the research community. Some of these factors have changed with advances in technology, while others, such as social factors, have remained relatively consistent.

One factor that has been noted consistently as a factor associated with a successful virtual team is trust (Jarvenpaa, et al., 1998; Jarvenpaa & Leidner, 1999; Sarker, Lau, & Sahay, 2001). One contributing factor to early trust in a virtual team is holding a face-to-face meeting in the early stages of team development to allow team members to get to know one another (Maznevski & Chudoba, 2000; Suchan & Hayzak, 2001). Developing trust early in the team building process is important for virtual teams because it facilitates better communication and equips them to more productively handle technical challenges and uncertainty as they arise (Jarvenpaa & Leidner, 1999; Lipnack & Stamps, 1997). The concept of swift trust developed from studies on virtual teams and suggests that teams develop a faster and more fleeting form of trust in order to accomplish the task at hand (Coppola, Hiltz, & Rotter, 2004; Iacono & Weisband, 1997; Jarvenpaa & Leidner, 1998). Developing the team trust necessary for success can often be one of the biggest challenges a virtual team

faces. Many organizations are not willing to fund initial face-to-face meetings; therefore, teams have to depend on regular communications or a team member's reputation for trustworthiness. Often it is difficult to quantify the level of trust for a team member that one has yet to meet (McDonough, et al., 2001).

Working on a virtual team is not an innate skill, and research has shown that training has positively impacted teams with virtual workers (Cascio, 2000; Tullar & Kaiser, 2000; Van Ryssen & Godar, 2000). Training has proven to be most effective when it is provided early in the team's development and is offered consistently within the team. Common outcomes of early training are increased trust between team members, greater commitment to team goals and decisions, and increased satisfaction for individuals participating on the team (Tan, Wei, Huang, & Ng, 2000; Warkentin & Beranek, 1999). Tan, Wei, Huang, and Ng (2000) and Warkentin and Beranek (1999) advocate training in team communications to improve outcomes, while others advocate mentoring as an effective training technique (Kayworth & Leidner, 2002; Suchan & Hayzak, 2001).

Team cohesion is another important factor in establishing an effective virtual team. Research suggests that building team cohesion is one of the biggest challenges a virtual team may face (Kayworth & Leidner, 2000; Kirkman, Rosen, Gibson, Tesluk , & McPherson, 2002). Early findings showed that face-to-face teams were initially better at decision making and appeared more cohesive, but midway through the project virtual teams began to show more signs of cohesion and productive decision making (L. Chidambaram & Bostrom, 1993; Laku Chidambaram, Bostrom, & Wynne, 1990-1991). Lind (1999) found that women adapted to virtual teams better than men, lending credence to gender diverse teams.

Good leadership on a virtual team has been positively associated with team success (Kayworth & Leidner, 2000; Lurey & Raisinghani, 2001). The topic of leadership in virtual teams has taken on many focus areas from leadership style (Lurey & Raisinghani, 2001) to leadership stability (Eveland & Bikson, 1987), and more recently communication cues

(Ashmore & Townsend, 2011). Kaboli, Tabari, and Kaboli (2006) argue that leadership should be defined differently in the virtual team setting. Different skills are critical to virtual leaders, such as their need to be sensitive to cultural norms of global team members. Yoo and Alavi (2004) investigated the traits of emergent leaders on virtual teams and found that they tended to initiate communication, schedule more meetings, and integrate the virtual team members.

The success factor that has received the most attention in the field of virtual teaming is good communication (Powell, et al., 2004). In the 43 articles reviewed by Powell, Piccoli, and Ives (2004), 24 of them focused their study on the way the virtual team communicated. Studies have shown that co-located teams are better communicators than their virtual counterparts (McDonough, et al., 2001), despite the fact that virtual teams communicate more often (Galegher & Kraut, 1994). While communication is important when working on a co-located team, virtual teams offer a new dimension because they involve so many mediums with which to communicate. Each media option varies in terms of the amount of nonverbal communication conveyed (J. B. Walther & Tidwell, 1995), sender-to-receiver delays (Montoya-Weiss, et al., 2001), and relationship building opportunities (J. Walther, 1995). Successful virtual teams have done a better job of selecting the most appropriate media option for communication (Maznevski & Chudoba, 2000). For example, a virtual team that has been more successful would use a richer and more expedient form of communication, such as a conference call, rather than e-mail to brainstorm ideas.

Global virtual teams face additional challenges due to language barriers, cultural differences, and temporal discrepancies (Jarvenpaa & Leidner, 1998). Some techniques that have been suggested to overcome these challenges include team building exercises and training (Kayworth & Leidner, 2000) and exploiting collaborative technologies (Kerber & Buono, 2004).

2b. Process

i) Software Development Processes

Software development, also known as software engineering, is defined by the Software Engineering Body of Knowledge (2004) as, "systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software" (p. 23) The field of software development emerged in the 1950s with operating systems and became increasingly popular in the 1960s and 1970s as software became a central focus in computing.

The first, and still the most popular, software development process is referred to as the waterfall model. The waterfall model advocates sequential phases of development with each stage completing before the next begins with a focus on structure. For example, all software designs are completed before the coding phase begins. This methodology was first introduced in the 1950s at a conference where a software development methodology for SAGE and has continued to be predominant software development methodology (Benington, 1956; Craig, 2003). Despite the popularity of waterfall development, it has continued to be criticized in the field for being process heavy and unresponsive to the inevitable changes that arise during software development projects (McConnell, 2004).

At the turn of the century the world of technology became increasingly inundated with requests for new features. This was particularly true for Web sites because society was becoming more dependent on Internet conveniences such as electronic mail, e-commerce, and real-time news updates. Product development teams needed a new way to respond quickly to these demands to stay competitive in the changing market. The solution came in

the form of agile development validating assertions by Whiteside and Bennett that software development needs to become more of an iterative process (Whiteside, Bennett, & Holzblatt, 1988).

Agile development, also commonly referred to as "iterative development" is the over-arching term for processes such as agile, lean, extreme programming, Scrum, and Rational Unified Process (C. Larman, 2004). Each of these versions of agile development have their own nuances, but they share the common goal of avoiding "a single pass sequential, document-driven, gated-step approach" (Craig Larman & Basili, 2003, p. 47).

Including usability specifications into iterative development processes was introduced in 2002 when iterative development was starting to gain popularity (Carroll & Rosson, 2002). Carroll and Rosson (2002) stated that usability designs should not be a static recommendation or design, but rather should evolve and refine throughout the development process. Larman and Basili (2003) claim that although iterative development has been the new buzz in the IT industry in recent years, it is not a new concept. Researchers at IBM's TJ Watson Research Center published the first research publication that described a process that had the most similarities to today's agile or iterative development process (Zurcher & Randell, 1968). This process was recommended to IBM executives in 1969 as "A model becomes the system" (Lehman & Belady, 1985). IBM did not start adopting the contemporary model of agile development until the turn of the present century.

What is agile development? What makes the agile development methodology and why is it getting the reputation as the superior approach to the waterfall methodology? Agile is an over-arching term that includes iterative approaches to software development that embrace the values of the Manifesto for Agile Software Development. According to Williams and

Cockburn (2003) and the evangelists who authored the "Manifesto for Agile Software

Development" (Beck et al., 2001), agile is a process for developing software that values:

> "Individuals and interactions over processes and tools, working software over
>
> comprehensive documentation, customer collaboration over contract negotiation,
>
> responding to change over following a plan" (p. 39).

Doshi and Doshi (2009) suggest that agile development goes beyond the methodology a team

uses to create software and creates a culture of employees with common values.

One clear difference between agile development and waterfall development is how

project phases are scheduled. Table 1 shows Nerur, Mahapatra, and Mangalaraj's (2005)

comparison of traditional and agile development methodologies.

Table 1: Nerur, Mahapatra, and Mangalaraj's (2005) traditional (waterfall) and agile comparison

	Traditional	Agile
Fundamental Assumptions	Systems are fully specifiable, predictable, and can be built through meticulous and extensive planning.	High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change.
Control	Process centric	People centric
Management Style	Command-and-control	Leadership-and-collaboration
Knowledge Management	Explicit	Tacit
Role Assignment	Individual—favors specialization	Self-organizing teams—encourages role interchangeability
Communication	Formal	Informal
Customer's Role	Important	Critical
Project Cycle	Guided by tasks or activities	Guided by product features
Development Model	Life cycle model (Waterfall, Spiral, or some variation)	The evolutionary-delivery model
Desired Organizational Form/Structure	Mechanistic (bureaucratic with high formalization)	Organic (flexible and participative encouraging cooperative social action)
Technology	No restriction	Favors object-oriented technology

In the waterfall methodology it is common for a project phase, such as the design phase, to take several months to brainstorm, design, review, modify, review, and finally approve the design. It is important to note that when approvals are secured, it means the design for the entire project is locked and a formal change request must be made to change the design after approval. A typical agile design phase is a matter of a week or two, but the design is not considered locked down until all project phases are complete (e.g. development, test) and the "iteration" or "sprint" is complete. Iterations or sprints are the time a project team is allotted to complete a particular portion of a product from start to finish. Figure 2 shows the phases that are included in a typical sprint. The graphic

highlights the fact that the iterative or agile development process allows usability professionals to be influential with the end product at all phases of the development lifecycle rather than just at the beginning with the waterfall method (Bittner, 2004; Jacko, Düchting, Zimmermann, & Nebe, 2007).

Figure 2: Iterative product development process (Bittner, 2004)

Another interesting difference is that the customers and stakeholders provide feedback at the end of each stage and designs are changed based on the feedback, which has proven to be extremely valuable in enhancing the usability of the product. This feedback cycle is often referred to as "participatory design or experiences" (Sanders, 2002). Team participation and interaction take place in what is referred to as a "Scrum" which is the managing body in an agile team that consists of a "Scrum Master" (team leader) and the other developers and

testers who are responsible for product delivery. Scrum teams typically meet daily to discuss priorities, goals, and customer feedback (Woodward, Surdeck, & Ganis, 2010). This cadence differs from the waterfall approach, which tends to hold weekly status meetings.

ii) Success Factors

Despite the general trend toward virtual teaming, iterative development teams strongly advocate that co-location of team members is a key success factor. The Agile Manifesto (Beck, et al., 2001) emphasizes this by stating, "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation." Co-location is not unique to iterative development and is regularly found with teams using waterfall methodologies; however, traditional teams tend not to report that co-location is critical to the success of their project. Co-location allows teams to use agile tools such as the information radiator and whiteboarding; but, more importantly, it helps to build trust and mutual understanding between team members (Cockburn, 2002). In development environments where co-location is not possible, some best practices have been identified, such as aligning the architecture of the product by geography (Coplien & Harrison, 2005), including and assigning ownership to team members (Woodward, et al., 2010), and using collaborative tools (Hunt, 2006).

Both waterfall and agile development approaches benefit from customer feedback (Ferreira, Noble, & Biddle, 2007; Gruner & Homburg, 2000). The difference between the two approaches is that traditional development advocates customer input before and after the product has been developed, whereas companies who have successfully implemented agile development processes have a customer-centric culture (Misra, Kumar, & Kumar, 2009),

where customers are providing feedback and offering suggestions throughout the entire development process. While some companies use stakeholders such as product managers as surrogate customers (Ramesh, Cao, Mohan, & Xu, 2006), most partnered with customers who provide regular feedback during development. It can be difficult for companies to find customers willing or able to devote the time needed to make iterative development a success (Nerur, et al., 2005), but those that do are more successful (Hanssen & Erlend F., 2006). The Extreme Programming approach to agile even encourages the customer to be located on-site with the developers (Fraser et al., 2004; Hunt, 2006). Cockburn (2002) contends that regular communication between the development team and the customer must be present to ensure the most benefit. Klein and Canditt (2008) also encourage customers to complete opinion polls during the process to help measure the business impact.

Traditional development teams operate within an hierarchical or matrixed environment (Nguyen, 2006). Such structures work within the traditional environment, but agile development shifts the development process to empower the development team to have more influence on the end product (Anderson et al., 2003), while the manager acts as more of a facilitator. Likewise, team members need to understand the impact of their autonomy on the project (Ramesh, et al., 2006). Successful agile teams have a management team that embraces this shift and encourages the team to exert their influence and self-organize (Hoda, Noble, & Marshall, 2010; Nerur, et al., 2005). The agile manager imposes less control, but provides a safe environment for rapid change and encourages trust and collaboration between team members (Augustine, Payne, Sencindiver, & Woodcock, 2005; Nerur, et al., 2005). Both approaches require a committed sponsor to set the direction and provide support for

both the team members and the managers in the organization (Drew Procaccino, Verner, Overmyer, & Darter, 2002).

Traditional and agile teams alike need to ensure that their team members have the training and education necessary to be successful in their respective development methodology (Shaw, 2000). Training can involve mentoring from team experts or coaches, and it can also be more formal training such as classes or instructional books (Cockburn, 2002; Coplien & Harrison, 2005). Paired programming is an agile technique that involves two programmers working together to write the code. The idea originated in Extreme Programming (Beck, 2000) and it encourages programmers to cross-train, exchange real-time ideas, and code faster with less defects (Succi & Marchesi, 2001). Large organizations transitioning from traditional methods to agile methods can benefit from piloting agile with smaller teams and projects, then having those teams coach other teams based on their experiences (Cockburn, 2002).

It is also important that the team is equipped with the tools best suited to the methodology they are using. Tools not only include development tools such as code repositories and test trackers, but also tools that allow teams to interact and communicate effectively. Cockburn (2002, 2004) has coined the terms "high-tech tools" for those that assist with the development of software such as automated build systems and "high-touch tools" for those that facilitate social and psychological needs such as e-mail or instant messaging. He emphasizes it is more important to have the right mix of high-tech and high-touch tools than to assume a tool that is the best fit for one team would also be the best fit for another. If, for example, a team is co-located, they may benefit more from face-to-face tools such as whiteboarding, while a distributed team may find a wiki or group chat program to be

more useful. Hunt (2006) advocates that it is important to know when the appropriate tool is needed and that team members should be properly trained on the tools their company requires so they can reap the most benefit. Woodward, Surdeck, and Ganis (2010) suggest that virtual whiteboards and multi-user chat servers help to remove barriers for virtual teams.

iii) Challenges

Neither traditional teams nor teams who have moved to iterative development are immune to the challenges inherent in a software development project. Both approaches have their unique challenges, as well as challenges which are pervasive to all software development projects.

One of the biggest challenges for software development teams is distributed team members (Damian & Zowghi, 2003). Many IT companies have made the move to distributed and global team members to leverage around-the-clock development, worldwide talent, and in some cases, lower resource costs (Prikladnicki, Audy, Luis, & Evaristo, 2003). Despite the possible benefits, Damian and Zowghi (2003) found that distributed software development teams find particular challenges with communication, knowledge management, time zones, and trust. Companies using iterative methodologies and those who are moving to such methodologies find virtual teaming particularly challenging due to the emphasis on co-location and face-to-face interactions. Time zone differences make daily Scrums difficult, less rigid rules make teams feel that they have less control over the process and quality, and the lack of cohesion within the team hinders trust building (Nerur, et al., 2005; Ramesh, et al., 2006). Nerur, Mahapatra, and Mangalaraj (2005) warn that it may take an organization

longer to move to an agile development methodology if many of their team members are virtual, and more so if the team consists of a global workforce.

Another major challenge for software development teams is creating a quality product. Coplien and Harrison (2005) contend that many of the quality issues in software stem from the fact that quality is not designed into the product. They state that quality is an afterthought that does not get attention until the team is in the testing or beta phase, when it is too late to make changes to the product. Some teams also lack a quality assurance professional who helps to keep the focus on quality and drives specific quality requirements throughout the development process. Generally, development teams must compromise schedule durations and endure additional costs to ensure a quality product is developed, which many companies cannot afford in the competitive IT environment (Harter, Krishnan, & Slaughter, 2000). Agile development has helped increased quality by introducing test-driven development (Cockburn, 2002; Woodward, et al., 2010) and by receiving feedback during iterations. Agile also imposes additional challenges for distributed teams who benefit from specific quality guidelines because agile values quality guidelines that are more malleable (Ramesh, et al., 2006).

Knowledge management is another aspect of software development that teams struggle to master. Rus and Lindvall (2002) identified five major areas where software teams are challenged with knowledge management; they include: 1) knowledge about new technologies, 2) accessing domain knowledge, 3) sharing knowledge about local policies and practices, 4) capturing knowledge and knowing who knows what, and 5) collaborating and sharing knowledge. Project tools reduce some of the challenges (Henninger, 1997); however,

global teams are still deeply impacted by knowledge management issues (Ramesh, et al., 2006), particularly those that are moving toward agile methodologies.

2c. Adaptive Structuration Theory

The concept of structuration as a theoretical model was originally developed by sociologist Anthony Giddens in the mid-1970s as "an ontological framework for the study of human social activities" (Bryant & Jary, 1991, p. 201). Giddens published extensively on structuration theory; some of his most seminal works include: *New Rules of Sociological Method* in 1976, *Central Problems in Social Theory* in 1979, and *The Constitution of Society: Outline of the Theory of Structuration* in 1984.

According to Giddens (1977), structuration is the investigation of how and why societal structures remain in some cases and dissolve in others. Structuration theory differed from previous sociological theories in that it argued that human behavior is not based on the actions of individuals or society as a whole, but rather the norms within unique structures. These structures, and the rules that govern them, are malleable and can be expected to change over time (Giddens, 1986). As an example, the structure of attending college exists today in a similar form to the way it did in the 1950s, but many of the norms have changed. Classes are often available online, notes are taken on a computer, and students can communicate through their cell phones. Basic rules such as professors teaching classes, students studying at the library, and grades being rewarded have remained the same. Students may behave by one set of rules when they are at college, and adapt to a different set of rules when they return to the structure that is their hometown.

Cohen (1989) suggests that the popularity of Structuration theory is based on the idea that it "provides an account of the constitution of social life, the generic qualities of the subject-matter with which social sciences at large are concerned" (p. 1). Others have suggested that Structuration theory has received significant attention because of the extensive

publications Giddens has written on the theory and the lack of concreteness that leaves it open to interpretation as well as criticism (Bryant & Jary, 1991; Stones, 2005). As with many seminal theories, Structuration theory has provided the groundwork for new theories. One theory in particular, Adaptive Structuration theory, has gained significant traction in the field of information technology.

Adaptive Structuration theory was initially introduced by DeSanctis and Poole in 1990 as a framework for studying the interaction of groups and organizations with information technology (DeSanctis & Poole, 1990). Such a framework brought a new lens with which to investigate information technology and change in organizations. Prior research had focused on the technology itself rather than the social aspects the technology introduced in a group or organization. Adaptive Structuration theory highlights that a group or organization's perceived utility and benefit of a technology drives its outcomes and future use. This perception also influences how the technology changes over time for the group or organization.

DeSanctis and Poole expanded on the Adaptive Structuration theory in 1994 when they published *Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory*. The key components to Adaptive Structuration theory are Giddens' (1977) model of structuration as well as Ollman's (1971) concept of appropriation. Together these concepts illustrate how the groups or organizations adapt the technology to meet their goals and requirements. DeSanctis and Poole's study used a group decision support system tool to illustrate how the group's social structure influenced the tool's use and evolution.

The group's adaption of the group decision support system was analyzed based on the Adaptive Structuration model which DeSanctis and Poole (1994) describe as "an interplay between advanced information technologies, social structures, and human interaction" (p. 125). The model contains seven major components that influence the adaption of an advanced technology. See Figure 3 for an illustration of the model.
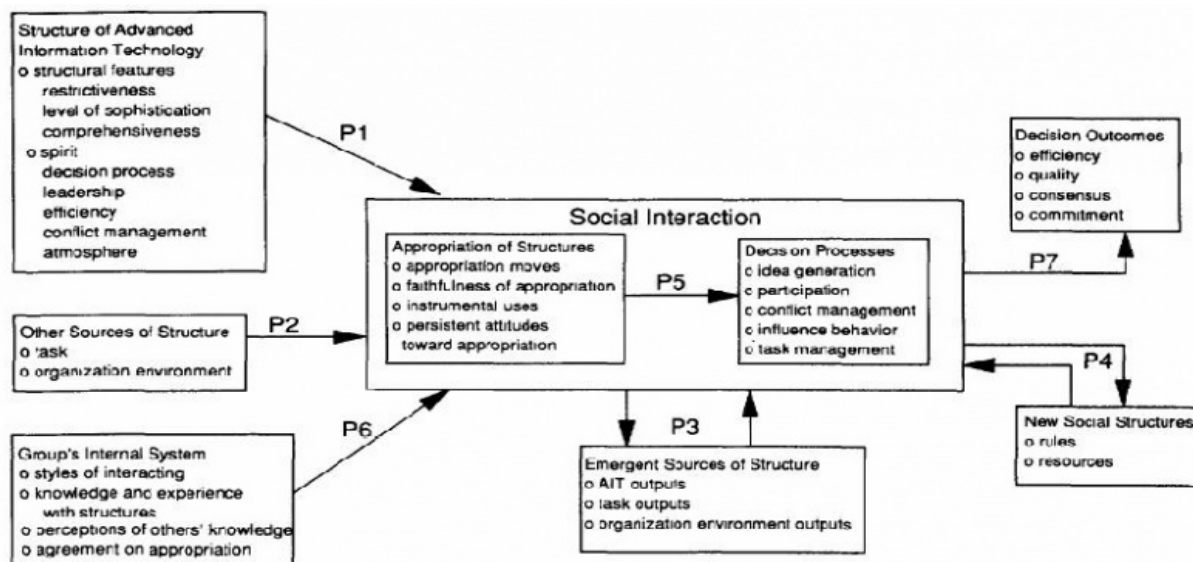
Figure 3: AST constructs and propositions

The first (P1) component is the structure of the advanced technology. These include the structural features, which are the technical capabilities of the advanced technology, and the spirit, which consists of the technology's metaphor, visual representations, and user assistance training. The second (P2) are any other sources of structure, such as the tasks that need to be completed by the users of the technology as well as the organizational environment that they are working within. Third (P3) are the new structures that emerge from the structure of advanced information technology (P1) and other sources of structure (P2). Fourth (P4) are the new social structures that result from the group or organization changing based on technology use or other outside factors such as new people joining the team. Fifth (P5) are the social interactions that occur as groups appropriate the new advanced technology and subsequently make decisions based on these appropriations. For example, a software development group may decide to adopt a new code repository, but decide that they are only going to put code for the new release in the new repository because they do not have the resources to migrate the old code. In time, this same team may decide to migrate the old

code to the new repository because maintaining two code repositories is too much of a burden on the team. Sixth (P6) are the group's internal systems which include the interactional norms of the group, the experience level of the team, the confidence they have in the knowledge of their teammates, and consensus on how best to appropriate the structure. The seventh (P7) and final component of the model includes the decision outcomes. The team or organization evaluates whether or not their appropriations and decisions have led to positive outcomes, such as greater efficiency or higher quality. Returning to the example of the new code repository, the team would evaluate if the new repository facilitated faster defect resolution and, ultimately, higher code quality in the product they are delivering. If the new repository resulted in more open complaints from customers, then they would need to look how they are using the tool and if adjustments may be needed in their processes and appropriations.

Adaptive Structuration theory has been used as a theoretical framework in both virtual teaming (e.g. Majchrzak, et al., 2000; Raghuram, et al., 2010) and software development research (e.g. Cao, et al., 2009; Ramesh, et al., 2006). In their comprehensive analysis of virtual teaming research Raghuram, Tuertscher, and Garud (2010) found that the virtual team cluster is commonly linked to Adaptive Structuration theory as a theoretical framework.

Maznevski and Chudoba (2000) investigated the dynamics and effectiveness of global virtual teams by using Adaptive Structuration theory to capture the categories and propositions for team analysis. They used four components of the Adaptive Structuration model as their major categories for the research design. These categories included structural characteristics (P1, P2), technology appropriation (P5), decision processes (P5), and decision outcomes (P7). Subcategories included some elements from the Adaptive Structuration model such as quality and task, but most subcategories were based on suggestions from other

previous research. The findings of their study generated the seven propositions listed in Table 2.

**Table 2: Maznevski and Chudoba's (2000) seven global virtual team propositions**

Proposition Number	Proposition
1	In effective global virtual teams, the higher the level of decision process served by an incident, the more rich the medium appropriated and the longer the incident's duration.
2	In effective global virtual teams, the more complex the message content of an incident, the more rich the medium appropriated and the longer the incident's duration.
3	In effective global virtual teams, if a rich medium is not required, the more accessible medium will be used.
4	In effective global virtual teams, if an incident serves multiple functions or messages, its medium and duration will be shaped by the highest function and most complexity.
5	A. In effective global virtual teams, the higher the task's required level of interdependence, the more communication incidents will be initiated. B. In effective global virtual teams, the more complex the task, the more complex the incidents' messages will be.
6	A. In effective global virtual teams, the greater the organizational and geographic boundaries spanned by the global virtual team's members and the greater the cultural and professional differences among team members, the more complex the team's messages will be. B. In effective global virtual teams, the stronger the shared view and relationships among global virtual team members, the less complex the team's messages will be. C. Other things being equal, in effective global virtual teams the receiving members' preferences and context determine the incident's medium.
7	Effective global virtual teams develop a rhythmic temporal pattern of interaction incidents, with the rhythm being defined by regular intensive face-to-face meetings devoted to higher level decision processes, complex messages, and relationship building.

Maznevski and Chudoba (2000) argue that the propositions generated from this study can be adapted to specific teams and organizations. They also contend that extending Adaptive Structuration theory to global virtual teams "provided the necessary descriptions of process and structure, of technology and social systems, and the interaction of these dimensions over time" (p. 489).

Majchrzak, Rice, Malhotra, King, and Ba (2000) applied the Adaptive Structuration model to a team developing a new product. They used the technology (P1), group (P2), and organizational environment structures (P2) to describe the pre-existing structures. From there they examined the appropriations of the collaborative technology used by the team during the initial phase of the project (P5) and again during the middle phase. Finally, they analyzed the reasons for the change in structures (P5) and did an assessment of the positive and negative outcomes of the project (P7). Their findings showed that initial structures were relatively consistent across the three organizations that were studied. They also found that misalignments in social structure and appropriation existed at the beginning of the project and continued after adaptations had been made immediately following changes in structure. Despite the misalignments, the team was still able to achieve project success, indicating that misalignments are not a sign of eminent failure on the part of the group or organization.

Cao, Mohan, Xu, and Ramesh (2009) used Adaptive Structuration theory as a framework for investigating how software development teams adapt to agile methodologies. They argued that the following four sources of structure influence the appropriation of teams to the agile development methodology: 1) agile methods defined through their structural features and spirit; 2) software project characteristics; 3) organizational context; and 4) each team's internal system that includes their interaction style, knowledge, and expertise with agile methods, and their perceptions about agile methods (p. 334). The research team analyzed interview transcriptions and code based on the structures found in the Adaptive

Structuration model. They continued consolidating like structure until the three dominant structures of Extreme Programming as a source of structure, other sources of structure, and the team's internal system were agreed upon. Four major appropriation practices were found: development process related, developer related, customer related, and organization/management related. The development process related appropriations included: abstraction in architectural design, design by formalized agreement, minimal traceability, post hoc documentation, and minimal documentation. The developer appropriations were paired for overlaps and empowerment through shared expertise, which emphasized the influence of the Extreme Programming agile approach. Customer related appropriations were a shared understanding of the specifications and an agreement on quality. The organization/management appropriations included upfront estimation and balanced formality. Figure 4 shows the modified Adaptive Structuration model for agile software development teams.
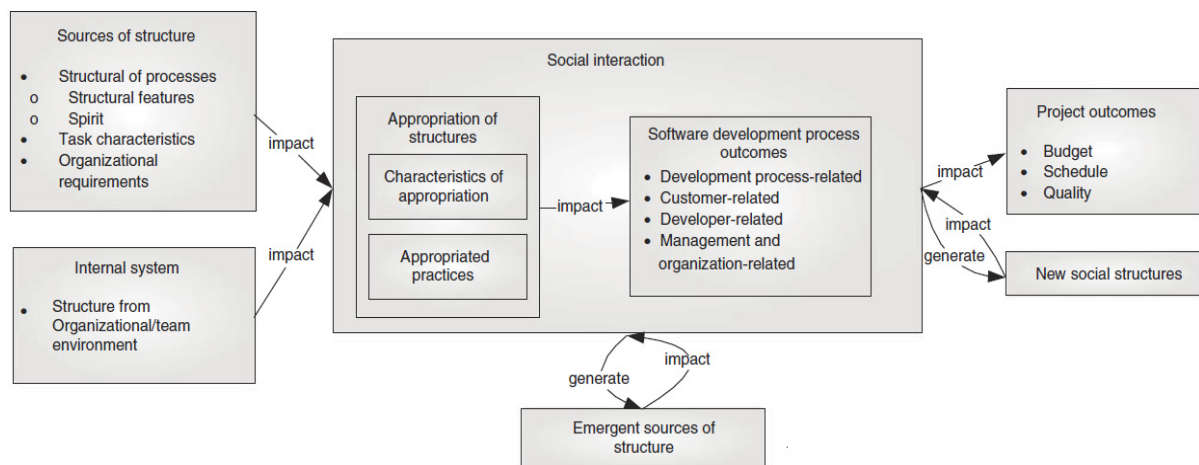
Figure 4: Cao, Mohan, Xu, and Ramesh (2009) AST model adapted for agile development

This study did not investigate adaption in terms of the appropriation of an advanced technology, but rather focused on the software development process (agile) as the adapted technology. The findings of this study add important new structures and appropriations that lend credibility to the idea that Adaptive Structuration theory can go beyond advanced

technology changes to adaptations in product development. The present study seeks to test the findings by Cao, Mohan, Xu, and Ramesh (2009) in a new context by comparing global virtual teams using waterfall and agile in the same organization.

2d. Propositions

Previous findings on virtual teams, software development processes, and Adaptive Structuration theory provide a theoretical basis for the likely differences that will be found between virtual teams using agile or waterfall software development methodologies. Cao, Mohan, Xu, and Ramesh (2009) adapted the AST model from seven to six components eliminating "other sources of structure" to make the model beneficial for the study of agile development teams. Findings by Cao et al. (2009) suggest that the following structures will be the most influential when waterfall teams move to the agile software development methodology: 1) agile methods defined through their structural features and spirit; 2) software project characteristics; 3) organizational context; and 4) each team's internal system that includes their interaction style, knowledge, and expertise with agile methods, and their perceptions about agile methods (p. 334). These findings suggest that:

Proposition 1: *Virtual teams using the agile methodology will demonstrate key differences from a waterfall team in the major structures found in the Cao, Mohan, Xu, and Ramesh (2009) model identified as the most influential while making waterfall to agile software development appropriations.*

DeSanctis and Poole (1994) define structural features as specific types of rules, resources, or capabilities that are part of the system. They define spirit as the values and goals of the structural feature. In this study the system is the software development process and spirit is the values defined in the Agile Manifesto (e.g. collaboration, trust, minimal documentation, embracing change, customer involvement). In their study of agile development adaptations, Cao, Mohan, Xu, and Ramesh (2009) define structural features as

the technologies, processes, and social action used by the team, while spirit is the general intent of the features of agile. Finding by Cao et al. (2009) argued that the structural features and spirit constructs under the "Sources of Structure" component were the among the most important differentiating constructs for agile teams in their AST framework adapted for agile teams.

1a. *An agile team will demonstrate structural features and spirit that are different than a waterfall team, given the differences in processes and values between the two methodologies.*

AST differentiates structural features from the organizational environment if the organizational environment provides the contextual structures that structural features are housed within (DeSanctis & Poole, 1994). Cao, Mohan, Xu, and Ramesh (2009) argue that the organizational context of an agile environment is one of decentralized decision-making and flattened organizational structures. Waterfall teams are usually hierarchical in structure and require that changes be approved through a very specific set of stakeholders (Raccoon, 1997).

1b. *An agile team will report that their structure is less hierarchical and their decision processes are less centralized than a waterfall team.*

An internal system in the AST model describes the nature of the members and their relationships inside the group (DeSanctis & Poole, 1994). Members of an agile team have an orientation toward collaboration, individual empowerment, trust, and knowledge-sharing that differs from the command and control nature of waterfall teams (Cockburn & Highsmith, 2001). Global virtual teams will adapt to a new internal system as they move from a waterfall to an agile methodology.

1c. *An agile team's internal system will be more collaborative than a waterfall team's internal system.*

AST defines attitude as the "the extent to which groups are confident and relaxed in their use of the technology process, the extent to which the group perceives the technology is of value to them, and their willingness to work hard and excel at using the system" (DeSanctis & Poole, 1994, p. 130). In the context of this study, attitude toward the development process rather than attitude toward a technology is used. Many teams moving to the agile development model have had some level of training and become more comfortable and confident with agile due to the regular use of retrospectives (lessons learned sessions held at the end of each project sprint). Waterfall teams may have had training in agile methodologies, but lack the experience to feel confident and relaxed with the process.

1d. An agile development team will have a more positive attitude about the agile development methodology than a waterfall team.

Doshi and Doshi (2009) found that moving to an agile development methodology does not just change the development process, it also changes the culture of the team. The Agile Manifesto, created by practitioners frustrated with waterfall methodologies, states that agile practitioners 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, " (Beck, et al., 2001, p. 1). The Agile Manifesto further emphasizes that such a fundamental shift in values can only be achieved by a dramatic change in the attitudes and dynamics of the development team. Maznevski and Chudoba's (2000) findings also suggest that cultural composition of the global virtual team (in this case the agile culture) is an influential structural characteristic.

Proposition 2: *The appropriations by an agile team will create a culture that is unique and distinct from a waterfall team.*

The Agile Manifesto that was created and advocated by the early adopters of agile development emphasized the importance of a collaborative environment when they listed "individuals and interactions over processes and tools . . . customer collaboration over

contract negotiation" (Beck, et al., 2001). Cockburn and Highsmith (2001) emphasize that agile team focus on individual competencies and increasing collaboration levels.

       2a. An agile team will describe their atmosphere as more collaborative than a waterfall team.

       Agile development requires that the team adopt new values and team dynamics (Beck, et al., 2001; Doshi & Doshi, 2009) that will shift the social interaction constructs of the team. Virtual team research has found that the richer the media used in the interaction, the more cohesive the feeling of team (Fiol & O'Connor, 2005), the more social presence the team member feels (Baker, 2002), and more trust develops between team members (Warkentin & Beranek, 1999). This suggests that a global virtual team using the agile development methodology will make more appropriations to media rich communication tools to overcome their inability to use co-located team members as the agile software development methodology advocates.

       2b. Technology appropriation decisions by an agile team will report a greater orientation toward technologies that support collaboration, team cohesion, and trust as compared to a waterfall team.

       Cockburn and Highsmith (2001) emphasize that an agile team empowers individual team members to ultimately decide the fate of the product. This includes working with customers and stakeholders to understand the most critical features or improvements that need to be incorporated into the product as well as the quality of the product that is released. Mnkandla and Dwolatzky (2006) argue that agile projects yield higher quality products.

       2c. An agile team will state that they have more control over the quality and outcome of the product they are creating than a waterfall team.

       Agile teams, specifically those using the Scrum and paired programming techniques, are encouraged to have regular interactions and most meet on a daily basis (Woodward, et al., 2010). One of the twelve principles in the Agile Manifesto specifically states that "Business

people and developers must work together daily throughout the project," (Beck, et al., 2001). There are not specific rules in the waterfall methodology that state that the team should not meet on a daily basis; however, waterfall teams traditionally meet less regularly and the interactions are more formal in nature (Sawyer, 2004).

2d. An agile team will have more interactions between team members than a waterfall team.

One important premise of agile development is that change is embraced rather than discouraged through trusting relationships, team empowerment, and a focus on customer satisfaction (Cockburn & Highsmith, 2001; Williams & Cockburn, 2003). Change is possible for teams using the waterfall methodology, but the process is more time intensive, formal, and is ultimately discouraged due to one cycle planning preferences (Raccoon, 1997).

2e. An agile team will be more likely to embrace project change and make appropriations that support change compared to a waterfall team.

CHAPTER 3. METHODOLOGY

3a. Context and Participants

i) Context

The sample selected for this study was drawn from a large global IT company. This company was selected because it has a long history of software development; it is cutting-edge in virtual and global teaming, and it has a strong desire to move from the waterfall development methodology to one that is agile and customer-focused. This company started deploying agile projects in 2005 and they have invested heavily in training their development organizations on agile best practices. Agile development was not required for all projects, but it was strongly encouraged. There was a good mix of products that were using the waterfall methodology only, using a hybrid of waterfall and agile methodologies, or had converted completely to an agile approach.

The decision to use the waterfall development methodology or move to the agile methodology was ultimately decided by the individual teams. Executive management encouraged and enabled the adoption of agile development, but entrusted the team to select the process that was best-suited for the project. This company deployed agile coaches to help their development organizations transition seamlessly to agile. These coaches worked with a wide variety of project teams and, therefore, brought a wealth of knowledge and best practices to the teams they were coaching. Agile training was available for all employees and, in many cases, was targeted toward specific job roles such as "Agile for Project Managers" or "Agile for Software Developers". Most of the training was available online, but occasionally courses were offered in-person at the major development labs.

ii) Participants

The waterfall and agile groups for this study are all employed by a global IT company that has been in the software development business for many years. They reside in the same organization, but work on global virtual teams on two distinct products. They report to the same director, but have different first and second-level managers. Team members did not self-select which project team they wanted to work on, but rather most had remained on the same team for many project cycles. Both teams are working on software products that have been on the market a number of years and have an established customer base. The waterfall team is working on a product that backs up and recovers critical data for organizations. The agile team is working on a product that monitors data centers and provides disaster recovery solutions. Both teams have a long history of using the waterfall approach to develop software and are working on projects of similar complexity. The defining difference between the two teams is the software development process appropriations they have recently deployed.

These two large globally-distributed teams using two different development approaches offer the ideal opportunity to study AST differences associated with the two software development approaches. This allowed the study to control for typical constraints, such as the differences between companies, the nuances that could occur when developing different types of products, or teams who are more virtually distributed than others. Despite the differences between individual personalities of team members, the teams only differed in the process (waterfall vs. agile) they were using to develop the product.

The first project team selected, the agile team, offered a good fit for this study because they were undertaking a large agile project. They were also the first team in their segment area of the company to move to agile. Extensive changes were needed in their team dynamics, tooling, processes, and mindset to make their transition possible. These team members put a lot of focus on documenting best practices and were motivated to participate

in this study because they wanted to understand how they could improve for future projects and share their experiences with other teams in their area who are moving to agile. This project took place over three years and was not yet completed by the conclusion of this study. This project had some pre-determined requirements, but the usability requirements had some room for interpretation and change based on customer and stakeholder feedback.

This team used a software product called Rational Team Concert (http://www01.ibm.com/software/rational/products/rtc/#), which was the recommended, but not required, project tracking tool at their company. Rational Team Concert (RTC) is a tool that brings together the tools required to deliver a product supportive of the agile methodology. The "team organization" feature was used for communication about user stories. This included feedback on user stories, approvals, and any changes that needed to be made. This team also made extensive use of "artifacts" in RTC, which allowed them to link the code repositories with project status for real-time tracking. This feature was also used to track code defects. One key requirement for this team was the ability to track earned value by showing the actual story points completed compared to the projected story points completed by sprint. RTC was deployed as a pilot tool for a smaller project by this team prior to the general deployment that was used during this study. Training on RTC was provided (by the firm) to the entire team before the start of the project.

In addition to RTC, the team used two other tools to manage the project. The first was Rational Quality Manager (http://www-01.ibm.com/software/awdtools/rqm/) to track test cases. The team felt that while RTC did offer a way to manage test cases, it was too divergent from the test tracking system they were accustomed to using. Rational Quality Manager (RQM) provided a more streamlined transition to agile for the test team. In addition, access-controlled databases were used to store controlled documents for the project, such as approved project plans and legal documents.

Members of this team had worked together on previous releases of this product, but they were all using the agile development methodology for the first time. This large software development team is divided into 12 distinct Scrum teams who are responsible for different features of the product. While some Scrum members are co-located, all Scrum teams have virtual team members and all Scrum teams have globally-distributed team members. A "Scrum of Scrums" was also held that included all of the Scrum Masters. All team members have worked on virtual software development projects prior to participating on this project. Each Scrum consists of male and female team members that range in experience from new hire to experienced professional.

Training on the agile development process was available for all of the team members, both in the form of on-site classes and virtual training programs. These training programs were available both before the project started and during the project life-cycle. The company did not require the employees to participate in the training programs; the result was that some, but not all team members, took advantage of the training that was available. In addition, a professional agile coach, who was also employed at the company, held four training programs with the project team to prepare them to transition to agile. These sessions were attended by many team members, but were not required. The agile coach continued to work with the team through the project by answering their questions and holding weekly sessions to help them through challenges and encourage best practices. About midway through the project, the company started to put more focus on encouraging all of the development teams to adopt the agile methodology. This push meant that the team members received more information through e-mail and on company wikis to help facilitate the use of agile. Company meetings were held to discuss the benefits of agile development and company policies and processes were modified to make agile development a more viable option for development teams.

The second team, the waterfall team, consisted of team members who have worked on previous iterations of the product together, and all of the projects have used a waterfall methodology. This team is comfortable working in the global virtual team environment together, and they have optimized their environment and processes based on lessons learned in previous projects. They communicate primarily using an instant messaging tool, e-mail, and teleconferences. Project status is typically reviewed on a weekly basis, with executive reviews on a monthly cadence. Their tools of choice are internally developed databases that contain key project information such as project plans, designs, and legal documents central to the project. They use the same code repository and test case tracker they have used in previous projects. The team consists of 268 team members worldwide. They have received extensive training on the waterfall software development methodology, and they are regularly audited by their parent company for process compliance. Their product backs up data to disk and tape drives to ensure organizations retain their required information for business and compliance purposes. Their product spans a much larger customer base and is the highest revenue driver for their organization. The nature of their product and their importance to their organization's bottom line mean that quality and reliability are critically important. At the time of the interviews they were completing a large project that spanned well over a year, but it was not the largest project they have worked on in recent years. User experience was an important aspect of this project; however, technical quality and reliability are more important factors for this release. Project requirements were pre-determined by market requirements and customer feedback from previous projects. This team has had limited training on the agile development methodology, but does have plans to change their development process to agile in the near future. Their organizational leaders have suggested that other teams try the agile methodology before this team makes the transition.

3b. Design

A qualitative field study approach was used in this research investigation. Dul and Hak define a case study as "one case (single case study) or a small number of cases (comparative case study) in the real life context are selected, and score obtained from these cases are analyzed in a qualitative manner" (2008, p. 4). Yin emphasizes that a good justification for using the case study methodology is to "understand a real-life phenomenon in depth, but such an understanding encompasses important contextual conditions – because they were highly pertinent to your phenomena of study" (2009, p. 18). The present case would be difficult to replicate in an experimental setting because much of the context that is central to virtual teaming, such as regular communication and overcoming real-life challenges, would be lost in a contrived environment.

The design of this study is based on DeSanctis and Poole's Adaptive Structuration theory (1990, 1994). DeSanctis and Poole's seminal article on Adaptive Structuration theory states that "documentation of a new structure formation will require longitudinal observation of the group and identification and persistent use of the technology-based structures in the group or organization at large," and advocates the use of written transcript and audio recordings to capture the words of team members for categorization (1994, p. 139). The case study approach is the best research approach for these requirements and has been used in similar studies (Cao, et al., 2009). Consistent with Cao, Mohan, Xu, and Ramesh (2009) use of Adaptive Structuration theory to study software development teams, this study uses software development process as the object of adaption rather than an advanced technology. While the present study does seek to understand how the use of technology differs between the two software development teams, it is only one aspect of the team's use of the software development processes and appropriations. Structures, appropriations, decisions, and decision outcomes will be based on the team's use of the waterfall and agile software development methodologies.

Eleven focus groups (five from the waterfall and six from the agile team) were conducted with the waterfall and agile development team members. Waterfall and agile team members were interviewed separately. Smaller focus groups were used to accommodate the schedules of the team members, but ultimately the responses were analyzed between two groups (n=2), the waterfall and the agile groups. Additional interviews with second line managers, project executives, and the Development Director were conducted to capture the leadership perspectives from the organization. One focus group included a member by phone because the participant was not able to travel to the interview site, but all of the other interviews were conducted in-person. Interview requests were sent to 77 team members located in the United States, and 44 agreed to participate. An e-mail was sent to the participants prior to the interviews asking them to complete the demographic information found in Appendix A.

3c. Procedure

One-hour interviews were conducted by the research team. Focus groups were separated by process type used (waterfall, agile, or mixed), then broken down again by similar roles in the organization. Developers, testers, team leads, Scrum masters, and technical writers were combined in focus groups. Project leaders, including first-level managers and project managers, were interviewed together. Second-level managers were interviewed together in the same focus group; executives were interviewed separately, and their feedback was used to supplement the information gathered in the focus groups. They were separated because they are not an active part of the day-to-day interactions of the virtual teams, and because many of them manage both waterfall and agile teams, which makes it difficult to identify their strongest team associations. Separating the focus groups by rank was prompted by concern that lower-level employees would be less open and honest about their interactions if they spoke in front of their managers or the executive team.

A set of interview questions was provided to team members prior to the interviews (see Appendix A for questions). This allowed team members to ponder responses in advance, and ensured they felt comfortable with the questions and had no reservations about participating in the interview. Interviews were not always kept entirely to script because follow-up and clarifying questions were often needed to understand the response and context. The interview questions were designed to help the researchers understand the business context and virtual team dynamics through the lens of Adaptive Structuration Theory. See Table 3 for a listing of how the interview questions map to the major sources of structure.

Yin (2009) and Fowler (1995) emphasized the importance of conducting a pilot study to refine the procedures and practice for the official interviews. The interview questions were piloted with a representative group of three team members in the software development organization from the company to check for clarity and appropriateness of the questions. Some of the questions were modified for clarity based on feedback from the pilot, but none of the questions were eliminated. Responses from the pilot were not included in the study's final results.

Table 3: Mapping of AST to interview

Constructs and Propositions	Definition from DeSanctis and Poole (1994)	Interview Questions

Structure of Advanced Information Technology (P1)
 	
• Structural features
  o Restrictiveness
  o Level of sophistication
  o Comprehensiveness
 	
• Spirit
  o Decision process
  o Leadership
  o Efficiency
  o Conflict management
  o Atmosphere
	Advanced information technologies provide social structures that can be described in terms of their features and spirit. To the extent that advanced information technologies vary in their spirit and structural feature sets, different forms of social interaction are encouraged by the technology.	• How much freedom do you have to modify your development process?
		• How well-established is your process (i.e., new, old, had many versions)?
		• Do you feel you have all of the information you need to work within your development process?
		• How are decisions made within the development process that you currently use (i.e., team vote, leader decides, depends on the situation)?
		• Describe your leadership structure (i.e., hierarchical, matrix).
		• Do you feel that your process is efficient? Why or why not?
		• How do you resolve conflicts on your project?
		• How would you describe the atmosphere of your project?

Table 3: Continued

Other Sources of Structure (P2) • Task • Organization environment	Use of advanced information technology structures may vary depending on the task, the environment, and other contingencies that offer alternative sources of social structures.	• How are tasks normally executed (i.e., team decides on important tasks and works on them as a group, project manager assigns tasks and team members disperse and work on their assignments)? • What is the organizational environment (i.e., positive, negative, fast-paced, collaborative)?
Emergent Sources of Structure (P3) • Advanced information technology outputs • Task outputs • Organization environment outputs	New sources of structure emerge as the technology, task, and environmental structures are applied during the course of social interaction.	• Do you see any changes in the products you create based on the software development process you are using? • Has your software development process changed the way you communicate in virtual teams? • Has the frequency of your communication with virtual team members changed?
New Social Structures (P4) • Rules • Resources	New social structures emerge in group interaction as the rules and resources of an advanced information technology are appropriated in a given context and then reproduced in a group interaction over time.	• How are rules created and modified? • How are resources allocated? This can include people and technology such as test machines or project repositories.

Table 3: Continued

Social Interaction (P5)
• Appropriation of structure
   o Appropriation moves
   o Faithfulness of appropriation
   o Instrumental uses
   o Persistent attitudes toward appropriation
• Decision processes
   o Idea generation
   o Participation
   o Conflict management
   o Influence behavior
   o Task management
	Group decision processes will vary depending on the nature of the advanced information technology appropriations.	• Do you feel that you use your software development process in its entirety or do you, for example, just use the parts that are the most useful to your team?
		• How faithful do you think your team is toward the software development process they are using (i.e., wikis, Lotus Notes, Rational Team Concert)?
		• How are these tools selected by your team?
		• Are there any tools you would like to use, but are not using and why?
		• What are the team's attitudes toward these tools?
		• How are ideas generated on the team?
		• How would you describe the participation of team members in all locations?
		• Do you feel that conflict is high, average, or low compared to other projects?
		• How do team members influence the project and other members of the team?
		• How are tasks typically managed (i.e., status collection, review meetings)?

Table 3: Continued

Group's Internal System (P6) ○ Styles of interacting ○ Knowledge and experience with structures ○ Perceptions of others' knowledge ○ Agreement on appropriation	The nature of the advanced information technology appropriations will vary depending on the group's internal system.	• What styles of interaction are typically used by the team (i.e., constructive, aggressive, passive)? • How much knowledge and experience does the team have with the software development process you are using? • Do team members have sufficient knowledge and experience to do what is asked? • How much agreement is there between team members on the process used?
Decision Outcomes (P7) • Efficiency • Quality • Consensus • Commitment	Given advanced information technology and other sources of social structure, and idea appropriation processes, and decision processes that fit the task at hand, then desired outcomes of advanced information technology will result.	• Do you feel your team has the ability to make efficient decisions? • Do you think quality decisions are made by your team? • Do you think consensus is typically achieved in your project? Why or why not? • Do you feel there is commitment from the team on decisions that are made? Why or why not?

The interviews were conducted using in-person focus groups. In-person interviews were held at two of the major development sites in conference rooms over a two-day period. Individual participants in the in-person focus groups were selected based on their availability to participate at the development site, role on the team, and, in some cases, based on the recommendation of the management team. Recording devices were used to capture

interview responses and copious notes were also taken during the interview to ensure accuracy and consistency with Adaptive Structuration theory approaches (e.g. Scott & DeSanctis, 1992). Project documents were also reviewed as needed to support or refute the information that was provided during the interviews.

Forty-four of the seventy-seven invited employees participated in the focus groups, yielding a 57% participation rate. Focus groups were held in conference rooms; group sizes ranged from one to six participants. Each interview ranged from 30 to 60 minutes, depending on the availability of participants. Responses to the demographic questions in Appendix A were requested of participants prior to the interviews to allow for a better understanding of the groups. They were able to e-mail their responses to the principal investigator prior to the interview or complete a paper copy at the actual interview.

Participants were sent a copy of the questions in Appendix A prior to the interviews and were given a chance to ask any clarifying questions prior to the start of the interview. Each focus group was given a brief description of the research study, notification that the interview was being recorded, and a final opportunity to ask questions. The interview format was semi-structured, with questions from Appendix A modified in some cases to provide clarification on responses from prior groups or to follow up on unexpected descriptions that were shared.

Once the interviews were completed, the recordings and interview notes were reviewed, transcribed, and analyzed using the process outlined in DeSanctis and Poole's Adaptive Structuration Theory (1994). Each interview transcription was first coded using NVivo (www.qsrinternational.com/products_**nvivo**.aspx*) by the primary investigator in accordance with the coding guidelines outlined in Appendix B. Two additional coders were recruited to validate the coding assignments. Each coder was given approximately a half-hour training on the Adaptive Structuration theory, the survey questions, the nodes, and

general training on the NVivo coding tool. One coder coded all of the transcriptions, while the other just coded file WS510019.dox to provide additional validation on the coding.

Each transcript was categorized as waterfall, agile, or mixed depending on the background of the participants in the interview. The mixed label was used when the participants were involved with both waterfall and agile projects. Ultimately, the interviews labeled as "mixed" were removed from the study due to the fact that they could not be categorized as agile or waterfall and biased the responses from each of these groups due to their experiences with both methodologies. Also, all of the participants in the mixed category were in executive management roles and had not been in the trenches of either an agile or waterfall project in recent years.

The inter-rater agreement percentage was calculated for all nodes in all transcripts. The average agreement between nodes was 97.09%. Table 4 shows the percentage agreement by node. Overall, the agreement for nodes in the waterfall interviews was 96.92% and the agile interviews had a 97.25% agreement. Krippendorff (2003) argued that an agreement percentage over 80% is considered valid, while scores from 67-79% are still considered acceptable. Other scholars, such as Riffe, Lacy, and Fico (2005) have questioned the validity of agreement scores below 70%.

Table 4: Comparing waterfall and agile interviews inter-rater reliability

Node	Waterfall (%)	Agile (%)	Average%
Advanced Information Technology Outputs	97.66	97.73	97.70
Agreement on Appropriation	100.00	99.38	99.69
Appropriation Moves	91.80	95.92	93.86
Atmosphere	96.17	95.23	95.70
Commitment	100.00	100.00	100.00
Comprehensiveness	97.47	95.44	96.46

Table 4: Continued

Conflict Management	98.50	99.11	98.81
Conflict Management 2	98.69	97.91	98.30
Consensus	99.30	99.81	99.56
Decision Process	96.60	98.22	97.41
Efficiency	97.38	97.62	97.50
Efficiency 2	100.00	99.90	99.95
Faithfulness of Appropriation	96.95	98.77	97.86
Idea Generation	99.40	99.87	99.64
Influence Behavior	99.83	99.45	99.64
Instrumental Uses	98.08	96.10	97.09
Knowledge and Experience with Structures	97.69	98.40	98.05
Leadership	97.11	95.84	96.48
Level of Sophistication	97.28	95.73	96.51
Organization Environment	79.31	91.99	85.65
Organization Environment Outputs	96.64	93.84	95.24
Participation	100.00	97.46	98.73
Perception of Others' Knowledge	97.47	97.99	97.73
Persistent Attitudes Toward Appropriation	88.06	96.92	92.49
Quality	99.59	99.72	99.66
Resources	99.29	99.42	99.36
Restrictiveness	98.32	96.35	97.34
Rules	99.44	97.52	98.48
Styles of Interacting	98.24	95.80	97.02

Table 4: Continued

Task	90.97	96.33	93.65
Task Management	95.00	93.30	94.15
Task Outputs	99.16	95.03	97.10
Total	96.92	97.25	97.09

Node coding assignments were reviewed and any phrases that did not have consistent coding were removed from the study. The responses associated with each node were then organized by the proposition that they were designed to support or refute. Nodes were associated with each proposition based on the portion of AST with which they were logically mapped. For example, Proposition 1a states that the agile team will demonstrate structural features and spirit that are unique to agile software development teams as compared to the waterfall team, given the differences in processes and values between the two methodologies. This proposition includes structural features and spirit from AST; therefore, the constructs associated with structural features and spirit (restrictiveness, level of sophistication, comprehensiveness, decision process, leadership, efficiency, conflict management, and atmosphere) are used for analysis.

Responses were summarized in a checklist matrix (Miles & Huberman, 1984) to understand the differences between the teams using the waterfall and agile software development methodologies. The checklist matrix format was used because it helps to highlight the key differences between the waterfall and agile teams to ultimately demonstrate how team process mediates virtual team interactions. A proposition was marked as "supported" if the majority of the responses within the selected constructs supported the proposition and "not supported" if the responses did not support the proposition. If a

construct had one to two responses that did not support the proposition, but the overall

responses still met the majority threshold, then it was noted as weakly supported.

CHAPTER 4. RESULTS

4a. Study Results

i) Demographic Data

A summary of the demographics of the participants can be found in Table 5 below.

Table 5: Demographic data

Question	Response (n=2 groups) Group 1: Waterfall participants = 22 Group 2: Agile participants = 19
Is your sex male or female?	**Waterfall Team:** Male 60% Female 40% **Agile Team:** Male 79% Female 21%
How long have you worked in the software development profession (in years)?	**Waterfall Team:** High: 35 years Low: 4 years Mean: 17 years **Agile Team:** High: 30 years Low: 3 years Mean: 12 years
How long have you worked at your company (in years)?	**Waterfall Team:** High: 37 years Low: 2 years Mean: 19 years **Agile Team:** High: 26 years Low: 3 years Mean: 12 years
How long have you worked on the specific team you are on now (in years)?	**Waterfall Team:** High: 17 years Low: 1 month Mean: 5 years **Agile Team:** High: 11 years Low: 7 months Mean: 5 years

Table 5: Continued

What is your role on the team?	Roles included: Developer, Tester, Project Manager, Development Manager, Test Manager, Information Development, Development Second Line Manager, Test Second Line Manager, Information Development Manager, Lab Support, Level 3 Support
How long have you worked with team members who are not at your site (in years)?	**Waterfall Team:** High: 42 years Low: 1 years Mean: 12 years **Agile Team:** High: 15 years Low: 1.5 years Mean: 8 years
How many team members do you communicate with on a regular basis?	**Waterfall Team:** High: 138 people Low: 5 people Mean: 33 people **Agile Team:** High: 130 people Low: 3 people Mean: 21 people
How many of these team members are not located at your site?	**Waterfall Team:** High: 100 people Low: 0 people Mean: 24 people **Agile Team:** High: 130 people Low: 1 person Mean: 17 people
Do you have team members that reside outside the US?	**Waterfall Team:** Yes: 99 % No: 1% **Agile Team:** Yes: 99% No: 1%

Table 5: Continued

How many team members do you communicate with regularly who reside outside the United States?	**Waterfall Team:** High: 100 people Low: 0 people Mean: 11 people **Agile Team:** High: 80 people Low: 0 people Mean: 7 people
Would you say your team is using a waterfall or agile software development methodology?	Waterfall: 54% Agile: 46%

ii) Proposition Results

Table 6: Proposition results

Proposition	Nodes Used in Analysis	Supported (yes/no/neither)
1a. *An agile team will demonstrate structural features and spirit that are different than a waterfall team, given the differences in processes and values between the two methodologies.*	• Restrictiveness • Level of sophistication • Comprehensiveness • Decision process • Leadership • Efficiency • Conflict management • Atmosphere	1a. Yes
1b. *An agile team will report that their structure is less hierarchical and their decision processes are less centralized than a waterfall team.*		1b. Yes
1c. *An agile team's internal system will be more collaborative than a waterfall team's internal system.*	• Styles of interacting • Knowledge and experience with structures • Perception of others' knowledge • Agreement on appropriation	Yes

Table 6: Continued

1d. An agile development team will have a more positive attitude about the agile development methodology than a waterfall team.	• Persistent attitude toward appropriation • Faithfulness of appropriation	Yes
2a. An agile team will describe their atmosphere as more collaborative than a waterfall team.	• Atmosphere • Idea generation • Participation • Consensus • Conflict management • Influence behavior • Styles of interacting	Yes
2b. Technology appropriation decisions by an agile team will report a greater orientation toward technologies that support collaboration, team cohesion, and trust as compared to a waterfall team.	• Instrumental uses • Advanced information technology outputs • Task outputs • Organizational environment outputs	Yes (weakly)
2c. An agile team will state that they have more control over the quality and outcome of the product they are creating than a waterfall team.	• Restrictiveness • Quality	Yes (weakly)
2d. An agile team will have more interactions between team members than a waterfall team.	• Participation • Idea generation • Task management • Organizational environment • Organizational environment outputs	2d. Yes
2e. An agile team will be more likely to embrace project change and make appropriations that support change compared to the waterfall team.		2e. Yes

Table 7: Waterfall vs. Agile Responses

Proposition	Waterfall Team	Agile Team
1a. *The agile team will demonstrate structural features and spirit that are unique to agile software development teams as compared to the waterfall team given the differences in processes and values between the two methodologies.* 1b. *The agile team will report that their structure is less hierarchical and their decision processes are less centralized than the waterfall team.*	• Restrictiveness – somewhat restrictive • Level of sophistication – very sophisticated • Comprehensiveness – very comprehensive • Decision process – team-based decisions made according to the process • Leadership – matrix and hierarchical leaning toward hierarchical • Efficiency – not efficient • Conflict management – escalation to management • Atmosphere – committed, focused, motivated	• Restrictiveness – somewhat restrictive • Level of sophistication – somewhat sophisticated • Comprehensiveness – somewhat comprehensive • Decision process – consensus by team, some decisions made by upper management • Leadership – matrix and hierarchical leaning toward matrix • Efficiency - efficient • Conflict management – resolve within the team • Atmosphere – collaborative, fun, positive
1c. *An agile team's internal system will differ from the waterfall team's internal system in that the agile team's internal system will be more collaborative.*	• Styles of interacting - situational, formal • Knowledge and experience with structures – many subject matter experts on the team • Perception of others' knowledge – skilled team members, experienced • Agreement on appropriation – mixed feelings about moving away from waterfall to agile, most are apprehensive about the change	• Styles of interacting - collaborative partnerships, accommodating, open communication • Knowledge and experience with structures – some knowledge and experience, still learning • Perception of others' knowledge – some experts, adjusted the team to help less skilled members, cross-trained team members • Agreement on appropriation – most team members have positive feelings about moving to agile

Table 7: Continued

1d. *An agile development team will have a more positive attitude about the agile development methodology than a waterfall team.*	• Persistent attitude toward appropriation – mixed feelings about making a change, some resistance to changing tools, some outright rejected any kind of move to agile, no clear vision of how it could be successful • Faithfulness of appropriation – make some moves, then revert back to old ways	• Persistent attitude toward appropriation – feel the move to agile was challenging but positive, most are happy to use the new tools required • Faithfulness of appropriation – embrace and continue to use new processes and tools unless directed by executives to do otherwise
2a. *An agile team will describe their atmosphere as more collaborative than a waterfall team.*	• Atmosphere - committed, focused, motivated • Idea generation – most projects ideas come from top down • Participation – active participation with team members (including global team members) • Consensus – general consensus among team members • Conflict management - escalation to management • Influence behavior – mandates from management, moves from competitive companies, maintaining quality • Styles of interacting – situational, formal	• Atmosphere - collaborative, fun, positive • Idea generation – a mix of top down and bottom up ideas • Participation – daily communication with team members (including global team members) • Consensus – strong consensus (including cross culture) • Conflict management - resolve within the team • Influence behavior – customer and stakeholder feedback, lessons learned • Styles of interacting – collaborative partnerships, accommodating, open communication

Table 7: Continued

2b. *Technology appropriation decisions by the agile team will report an orientation toward technologies that support collaboration, team cohesion, and trust than the waterfall team.*	• Instrumental uses – CMVC code repository, wikis to store project information, accept top down tool recommendation, have done video-conferencing & have moved away from that technology, e-mail, phone, Excel • Advanced information technology outputs – product is predictable, maintain same tools they have used for years • Task outputs – product designed before it is developed, product is industry rather than customer-driven • Organizational environment outputs - do not feel like underlying processes need to change, do not organize teams to encourage optimal communication	• Instrumental uses – wikis to communicate (including a social wiki), RTC for communicating, Lotus Notes, instant messaging, telephone, RQM, e-mail, team is more active in recommending new tools to management; Lotus Live for product demonstrations • Advanced information technology outputs – better product, functionality in product based on user stories, move to agile-friendly tools such as RTC, stakeholders involved in product reviews • Task outputs – shorter cycles, customer-focused, better products, more focused, efficient changes can be made • Organizational environment outputs - arrange teams for more communication and more co-location (teams and leaders)
2c. *An agile team will state that they have more control over the quality and outcome of the product they are creating than the waterfall team.*	• Restrictiveness – somewhat restrictive; deliver what is on the roadmap • Quality – quality focus	• Restrictiveness – somewhat restrictive; changed sprint lengths and some content • Quality – much improved over waterfall; defects found earlier

Table 7: Continued

2d. *An agile team will have more interactions between team members than a waterfall team.*	• Participation – less communication than agile, weekly meetings • Idea Generation – suggestions come top down • Task Management – weekly meetings and status updates, earned value, all plans made up front • Organizational environment – supportive, high stress, fast-paced, work and problem-solving focused • Organizational environment outputs – do not feel like underlying processes need to change, do not organize teams to encourage optimal communication	• Participation – more communication, daily meetings • Idea Generation – ideas are generated by the team based on customer feedback • Task Management – based on lessons from previous sprints, burn down charts, leaders must trust team members; less documentation, more contingency required • Organizational environment – collaborative, fast-paced, positive, flexible, customer-focused • Organizational environment outputs – arrange teams for more communication and more co-location (teams and leaders)
2e. *An agile team will be more likely to embrace project change and make appropriations that support change as compared to a waterfall team.*		

CHAPTER 5. DISCUSSION

5a. Analysis

i) Proposition 1

The results showed that Proposition 1: "Virtual teams using the agile methodology will demonstrate key differences from a waterfall team in the major structures found in the Cao, Mohan, Xu, and Ramesh (2009) model identified as the most influential while making waterfall to agile software development appropriations," was supported. Propositions 1a through 1d captured the structures within Cao et al.'s (2009) model that most influence the appropriation of teams using the agile development methodology. These structures include: 1) agile methods defined through their structural features and spirit; 2) software project

characteristics; 3) organizational context; and 4) each team's internal system that includes their interaction style, knowledge, and expertise with agile methods, and their perceptions about agile methods (Cao, et al., 2009, p. 334).

Differences were found between the waterfall and agile teams in terms of structural features and spirit, supporting Propositions 1a and 1b. Waterfall teams were more sophisticated, comprehensive, and made decisions in accordance with the development process they were following. The waterfall team leaned more toward a hierarchical leadership structure, as evidenced by the fact that they resolve conflicts through an escalation process, although they reported that they are technically organized in a matrix. One participant described the hierarchical environment as,

> "It is hierarchical, the way we work. So we may all report to the same director, and then there are second lines under them, and then the first lines are our people. Our people take directions from any one of those in the hierarchy. So, even though they report directly to the manager, they do not necessarily take directions from that manager. They'll take directions from release manager, which is the person who owns the products deliverable at that time."

Although the waterfall team did not feel that their process was efficient, they reported that their team atmosphere was one of commitment, motivation, and focus. In many ways, the waterfall team had a similar structure and spirit to a military organization. The agile team reported less sophistication and comprehensiveness, and made their decisions by team consensus. The agile team did, however, note that some decisions were still reserved for upper management. This raises the possibility that team structures are based as much on their corporate norms as their process choice. It may reflect the preferences of their managers and

team leaders rather than the processes they are working within. It is also possible that their roots in the waterfall method may still be influencing their behaviors. A team that only has experience using agile methodologies may report more team-based decisions. The agile team reported having a matrix structure that worked in an efficient manner, with conflicts typically resolved within the team. An agile team member stated that, ". . . there is a point that we all get together and identify how that actually resolves from someone who has an expertise in that area. We are trying to contain this within the team as much as possible." They felt their atmosphere had improved from the days when the waterfall process was used and is now more collaborative, fun, and positive. An agile team member described their relationship with their team as,

> "We all get along most of the time. I would say within the team it's a lot of fun, I think for me I am having more fun on the team than I have in a long time. It really feels in an agile way, we have to control the flexibility to do what we need to do."

One unexpected finding was that the groups did not differ in their feelings about the restrictiveness of their process. While almost all of the responses were consistent with the differences between the literature on waterfall and agile teams, this was one notable difference. The restrictiveness reported by both teams may be a reflection of the policies required to run a large and well-established information technology company. The company may be open to using new approaches, but they are not able to ignore legal and cross-organizational requirements, such as open source and globalization, that may be less applicable to smaller companies deploying the agile development process. This could also be due to both teams having roots in waterfall methodologies. Further research is needed using

teams from different companies to better understand differences in restrictiveness between waterfall and agile teams.

Proposition 1c argued that the internal system of an agile team would differ from the internal system of a waterfall team. The results show that this proposition was supported. The constructs included in the internal system are: styles of interacting, knowledge and experience with the structures, perception of others' knowledge, and agreement on appropriation.

The waterfall team was more formal, and prescriptive actions were taken under specific circumstances. For example, one waterfall participant said that, "There are a number of different levels that we work through. So, for a release, the release leads the release managers and the project managers meet weekly and discuss things that are needed to be done at that point of time and talk about issues." These differences could also be attributed to differing leadership styles of the first and second line managers on the respective teams and the nature of the products such that one backs up critical data and the other simply reports on the storage environment. Further research is needed to control for these possibilities. The waterfall team had been using the waterfall process for many years and felt that they were subject matter experts on the waterfall process. Similarly, they felt that their team members were competent in their ability to use the waterfall process due to their extensive experience. The waterfall team had generally negative feelings about moving away from the waterfall process, but there were a few optimists in the group. One waterfall participant cautioned that,

"I think from the test perspective for the agile, we are kind of worried about how many resources this is will take from us because our team is more individualized. We

used to have people coming from multiple lines. Are we still able to do it or will we have more people, or will we know we are not going to get it? There is a challenge to figure it out."

Another waterfall participant had a more optimistic perspective about moving to agile and stated that, " . . . we wanted to be more Agile-like."

The agile team had a different interaction style that valued collaborative partnerships, open communication, and accommodated the needs of the team. An agile team member described their team interactions as, "We do work really well and talk frequently, and especially when with the local ones [team members]. Most everyone wants to step up and help out a person that is struggling; that is a great attribute. You know, a team that is willing to help out." In terms of the team's knowledge, experience, and perception of others' knowledge, the agile team emphasized that they have some expertise with agile and are still in the learning phase. They understand that others on the team may have limited experience with agile and actively try to support team members who are still learning. Contrary to the waterfall team, the agile team had consistently positive feelings about the move to agile development. One agile team member stated that "It has gone healthier with agile than waterfall because in waterfall the last one or two months were like you were on a death march trying to finish up the project."

Proposition 1d predicted that an agile team would have a more positive attitude about the agile development methodology than the waterfall team, and the data supported this prediction. The two constructs, persistent attitude toward appropriation and faithfulness of appropriation, were reviewed to derive the attitudes of the teams.

The persistent attitude about moving to a more agile development model from the waterfall team was that it would negatively impact both their product and their team. They did not feel that they had a clear vision of how agile could be successful in their environment. They noted that they had tried some of the tools that their company endorses for agile projects, and these tools had taken a lot of time to implement and learn. The waterfall team said they were able to understand where agile might benefit a team that had different dynamics than them, such as a small team working on a new product that has co-located team members. In terms of faithfulness of the appropriation, the waterfall team had made some moves toward agile in the past, then reverted back to the traditional waterfall methodologies. When asked why the team reverted back, one waterfall team member explained, "I believe that given limited knowledge in agile, locality, size of the team, maybe the product, spread out, people not really realizing or sticking to the agile process. People are ingrained with waterfall, and it [agile] actually slowed us down a lot more until we gave up and said okay we are going back to waterfall." Their concern about moving to the agile process could also be a result of resistance from their management team, or possibly their team's desire to remove as much risk as possible from their project. On average, the waterfall team has been with their company and in the software development industry longer and as a result may be more resistant to change in general.

While the agile team also had a background in waterfall development, they had a very different attitude about agile once they had some experience using agile development. The agile team admitted that there were some challenges they had to overcome initially with time zone differences, new tools, and a new way of thinking and operating, stating that, "The agile process is a lot different than waterfall development and the way you structure the work, the

way you break down the things, the way you communicate with the team is lot different. We have to learn all this. It took time." They felt that the transition was worth it, reporting improvements such as, "I think the best part [of moving to agile], which I like, is the demonstration, where you demo all your hard work on Friday, every Friday. I think I like that a lot," and "I think the interaction has increased, quite a bit, significantly. And, I think that's been a benefit." The agile team also felt that the additional work of moving to new tools that supported the agile process was worth the effort. One team member reported, "There is a very good tool – Rational Team Concert. It is a very excellent tool that helped a lot because developers, testers, and management can all go and look at different views." If team members had self-selected either the waterfall or agile team, then concluding that personal attributes of the team member would lead them to choose either waterfall or agile. Given that team members were not able to choose, it seems more feasible that the process created the attitude change of the team members.

ii) Proposition 2

Proposition 2 argues that the appropriations by the agile team will create a culture that is unique and separate from the waterfall team. The data showed that the agile team had a distinct culture that was different from that of the waterfall team. These findings support previous research by Doshi and Doshi (2009) that agile development changes the culture of the team. The results also support findings by Maznevski and Chudoba (2000) that indicate global virtual teams also influence the culture.

Proposition 2a predicts that an agile team will describe their atmosphere as more collaborative than a waterfall team. The data confirmed that the agile team reported a more collaborative atmosphere than the waterfall team.

The waterfall team reported that their team members are committed to the project, focused on meeting project objectives, and motivated to do what the team needs to be successful. They also reported that team members support one another when assistance is needed; and they were also comfortable with one another, given the long history they have working together. One waterfall team member described it as, "It's a pretty well established team and I would say most of them are highly motivated to make things happen. They seem to put in whatever it takes to make it work, so they're committed to the product and the company."

The agile team shared a commitment to project success, which is a core value for their company as a whole, but put more focus on the relational aspects of the project team. They used the term collaborative more often and used more enthusiastic language when describing agile development and their team. An agile team member said of agile, " . . . there is a real bond while we are working together." Another said, "I say it's definitely a fast-paced and collaborative development team." Several agile team members emphasized that employees who were able to work together locally were able to create the strongest bonds between team members.

Proposition 2b focuses on the tooling decisions made by the development teams and argues that technology appropriation decisions by an agile team will report a greater orientation toward technologies that support collaboration, team cohesion, and trust as compared to a waterfall team. The data supported this proposition, although weakly. In some

cases the teams were empowered to make tooling decisions, while in other cases these decisions were highly encouraged or mandated by upper management or the corporation as a whole. When asked by the interviewer if they could find another tool that fits what you need more, one participant replied, "No. It has been driven by corporate management." Waterfall and agile teams who work for different companies, and who are given complete control to decide on the tools that are best for their teams, may make different technology choices.

In general, both teams used many of the same tools within their project teams. This is likely due to the fact that they both work for the same company and because the company does dictate many of the tools they must use. Both teams regularly used the phone, e-mail, instant messaging, screen sharing, and wikis. The waterfall team had tried to use video conferencing, but came to the conclusion that it was not beneficial. The primary difference between the two teams is that the waterfall team was resistant to moving to new tools, while the agile team embraced, and even pursued, the change to new tools. The agile team made a concerted effort to use more agile-friendly tools. They took the initiative to be early adopters of these tools, sought training, and exploited the collaborative features that the tools offered. In addition, the agile team also maintained a social wiki that contained a weekly newsletter with articles about team members to encourage team bonding.

Proposition 2c proposes that agile teams will feel that they have more control over the quality and the outcome of the product that they are creating than the waterfall team. The data supported this proposition, but again weakly. The minimal differences may be due to the fact that both teams are required to follow a corporate-wide quality process. Both teams reported that they felt somewhat restricted in the development process they were using, which again is likely due to the fact that they are both working within the same organization. The agile team noted they were able to gain support from their management team to change the duration of their sprints. They felt changing the sprint duration would ultimately benefit the end product. Both teams also felt that they had a quality focus, but the agile team felt that

product quality had improved in agile, primarily due to the fact that defects were being found much earlier in the development cycle. When asked about quality, one agile team member summarized the benefits of agile as,

"... flexibility in delivering the content that our stakeholders are really looking for and making sure that it's to their specifications and expectations, that we're meeting their performance expectations, that we're meeting a wealth of different things for them, in addition to improving our quality, the more we can shift left. The way we can catch things earlier, get that earlier when the code is wet, when the code is fresh and it's easier to make those tweaks and changes, we feel that means later on we won't have to suffer our big backlog of problems that we typically see in a waterfall project."

The agile team also felt confident they were creating a product based on the needs of the customer, whereas the waterfall team could not be as reactive to customer requests and focused on delivering the product that the company roadmap requested of them. One waterfall team member described it as, "you get a very large base of information and you can know it so far ahead that you can say that over the next two years we're going to do these four big things and we're not going to deviate much from that."

Proposition 2d predicts that an agile team will have more interactions than a waterfall team and the data supported that prediction. The waterfall team held weekly meetings to review project status in a formal manner using charts and assigned presenters. Team members interacted one-on-one or in smaller groups, but those interactions tended to be only as needed. When asked about team interactions, one waterfall team member said, "Just from the high level, we have a weekly status meeting. You know, so the leads here have a status meeting with the Beijing once in a week. We mostly have the leads in Beijing talk about what's going on. We do send e-mails to each other, but team members mostly interact with the team leads in other geographies." Consistent with the values of the Agile Manifesto, the agile team focused heavily on team interactions. Their use of sprints to manage project

cycles keeps the team working at a fast pace and daily Scrum meetings are a key part of their success. They reorganized the global team members so the work teams could be located in similar time zones. They reported that their decision to group the Scrum teams by time zone was a direct response to feedback during a retrospective that communication across time zones was a challenge. Team members still regularly communicate with other team members in different time zones, but the core team members they need to interact with are available during the same or at least similar work hours. One agile team member described the increased interaction with agile as, "But, in the Scrum meetings that interaction has increased. That has been very beneficial because we know exactly what the other person is doing." Almost all of the agile team members reported that the increased interaction with their team members positively impacted the overall project.

The final proposition, 2e, states that agile teams will be more likely to embrace project change and make appropriations that support change compared to the waterfall team. This proposition was also supported by the data.

The waterfall team was not completely adverse to change, but they were more resistant to making changes than the agile team. They felt that they had a solid, repeatable process that worked for them, and process changes introduced unnecessary risk to the project. One waterfall team member stated that, "Most of the time we try to live within the process. We don't try to change the process because it's so well established. If there is a real need to change the process, we'd go up to the process gurus that we have." The waterfall team also avoided changing tools, and indicated that changing to a new tool would be a major investment to the team in terms of training, moving project code and data, and creating new templates. They admitted they were aware there are tools available that could help their project, but they did not feel that changing to these tools was worth the risk. They did not make a process or tooling change unless it was dictated by upper management, and even when it was required, they reported that they would often seek an exception. Again, it is

possible that the waterfall team's greater average experience at the company and in the industry may influence their resistance to change.

The agile team had a different attitude toward change. They reported that change had generated extra work for their team in terms of training, tool migration, and adjusting to new interaction styles, but they felt short-term setbacks were worth it for the overall benefit of the product. The agile team had been one of the first in their area to try agile development, and they evangelized the benefits of agile to their management team so they could continue to use it in a more robust way with each release. They took the initiative to try new tools that were more supportive of the agile methodology. One agile team member noted, "So we started using RTC (Rational Team Concert) and in my mind is one of the best tools out there we could have adopted. I think we are finally there and have everyone on board, I think it is a big payoff for us." Another team member noted, "RTC collates data like nobody's business, but I think they'd have to do that on a sprint by sprint basis because we've seen so many changes between Scrum team make up just from sprint to sprint." The agile team secured an agile coach during their first year of transition to ensure they were using the process correctly. The coach also served as a resource that could address questions and concerns with the team on a regular basis. This coach was also available to the waterfall team, but they chose not to work with him. After each sprint the team held retrospectives or lessons learned sessions, and adjusted their process and team operations based on the feedback.

Table 8: Proposition results summary

Proposition	Supported (yes/no/neither)
1. Virtual teams using the agile methodology will demonstrate key differences from a waterfall team in the major structures found in the Cao, Mohan, Xu, and Ramesh (2009) model identified as the most influential while making waterfall to agile software development appropriations.	Yes
1a. An agile team will demonstrate structural features and spirit that are different than a waterfall team, given the differences in processes and values between the two methodologies.	Yes
1b. An agile team will report that their structure is less hierarchical and their decision processes are less centralized than a waterfall team.	Yes
1c. An agile team's internal system will be more collaborative than a waterfall team's internal system.	Yes
1d. An agile development team will have a more positive attitude about the agile development methodology than a waterfall team.	Yes
2. The appropriations by an agile team will create a culture that is unique and distinct from a waterfall team.	Yes
2a. An agile team will describe their atmosphere as more collaborative than a waterfall team.	Yes
2b. Technology appropriation decisions by an agile team will report a greater orientation toward technologies that support collaboration, team cohesion, and trust as compared to a waterfall team.	Yes
2c. An agile team will state that they have more control over the quality and outcome of the product they are creating than a waterfall team.	Yes
2d. An agile team will have more interactions between team members than a waterfall team.	Yes
2e. An agile team will be more likely to embrace project change and make appropriations that support change compared to a waterfall team.	Yes

5b. Limitations and Implications for Future Research

This study addresses some of the methodological limitations in previous virtual team research by comparing demographically similar virtual teams employed by an IT organization rather than creating lab-induced student teams.

Due to the case study approach used in this study, all of the virtual team participants were from the same organization. Using participants from different organizations will help to uncover if there would be additional differences between waterfall and agile team responses if they are not operating under the same organizational norms, rules, and values. The similarities in restrictiveness and control over quality and product outcomes in this study are likely the result of the employees sharing a common perspective about their organization in general. Further research is needed using virtual teams in different organizations using waterfall and agile methodologies. Research shows that waterfall and agile methodologies are sometimes deployed differently between small and large organizations (Kahkonen, 2004; Lindvall et al., 2004), so including teams of various sizes should be investigated. The agile team in the present study also had a long history of using the waterfall methodology. Studying an agile team that does not have extensive experience with waterfall may also highlight additional differences from a waterfall team. Care should be given to compare organizations that have less-experienced teams as well as experienced teams to better understand if their experience level impacts their attitudes toward change.

Similarly, studies have shown differences in global and non-global virtual teams (Jarvenpaa & Leidner, 1999; Maznevski & Chudoba, 2000), emphasizing the importance that both types of team should be included in future studies. The present study is limited to only global virtual teams. Finally, all-female virtual teams have shown to adapt differently to

virtual teaming (Lind, 1999), warranting studies that compare process appropriations of all-female and all-male teams.

This study is also limited to the investigation of software development processes. Virtual teams may adapt other types of processes, such as those used by non-profit organizations or those used by government organizations; further studies are needed to understand how representative software development processes are compared to the various types of project management processes that are available.

5c. Conclusions and Propositions for Future Research

The findings from the study highlight the differences in team dynamics between waterfall and agile teams and emphasize the impact of process choice on virtual teams. A software development team moving to agile will need to consider cultural, tooling, and attitude shifts that need to occur to make the transition. These changes are intensified when the organization is using virtual teams. The virtual team members may have to change their structure to support daily interactions and organize around similar time zones. The organization also needs to be prepared for more regular change. This includes adapting to new tools that support the fast-paced nature of agile development as well as the regular communication needed to work collaboratively. Given the challenges with appropriating a global virtual team to agile, organizations may trend toward more co-located teams that can collaborate more efficiently. Agile research has shown that co-located teams are the most ideal structure for agile development (Law & Ho, 2004). Previous research and the findings from this study would suggest that a co-located team would appropriate to the agile methodology faster than a virtual team.

This research also builds on previous virtual teaming research by demonstrating that the process, or more generally, the organizational rules that a virtual team is following will

impact the team interactions. This confirms Martins, Gilson, & Maynard's finding that task type, social context, and time mediates virtual teams (2004). Virtual team members will not necessarily interact because their technology allows them to do so or because they have been assigned to the same team. The framework they are working within will guide the amount of interactions and the nature of these interactions. Waterfall teams interacted less and the communications were more formal simply because they had used the waterfall process. A sister team who decided to use agile interacted daily, created core teams in similar time zones, and sought tools that would facilitate collaboration with virtual team members. These findings suggest that if a waterfall or agile team is compared to a virtual team using a different development process, such as rapid application development, each team would have unique virtual teaming dynamics to support their respective process. Process choice is an important factor to consider when managing and working within a virtual team.

This study also confirms the findings by Cao, Mohan, Xu, and Ramesh (2009) that there are unique constructs that need to be considered when adapting to an agile development model. Their AST framework was designed for agile development teams and their study only analyzed agile development teams. This study extends the use of their model to analyze both waterfall and agile teams, providing support that Cao et al.'s (2009) framework can be extended to aid in the analysis of software development environments

Gartner research predicts that 80% of software development projects will be executed using an agile development process by the end of 2012 (Murphy et al., 2009). Furthermore, research by the Project Management Institute (PMI) validates this trend when they found that the use of agile methodologies has tripled from December of 2008 to May of 2011 (2011). These predictions, combined with earlier predictions by Gartner of the pervasiveness of virtual teaming (Gartner, 2006), emphasize the need for a better understanding of how global virtual software development teams differ when they use different development methodologies.

They key to understanding how to create the best possible virtual development team is to have a sound understanding of the differences between the waterfall and agile teams. This study focuses on the organizational, cultural, and technological differences between a waterfall development team and an agile development team. The results emphasize that an organization must understand the differences between virtual teams using waterfall and agile development methodologies and cannot simply change their tools or the structure of their teams if they want to transition from one process to another.

This study makes the following contributions to research in the fields of virtual teaming, software development, and AST:

- The process a team deploys does mediate the dynamics of a virtual team.

- The majority of software development research studies either waterfall or agile development teams. This study compares the differences between the two teams in terms of organizational, cultural, and technological differences. Findings suggest that there are important differences between the two teams that need to be considered by team members and their leadership teams.

APPENDIX A. INTERVIEW QUESTIONS

Group #: _____

Demographic/Context:

1. Is your sex male or female?

2. How long have you worked in the software development profession?

3. How long have you worked at your company?

4. How long have you worked on the specific team you are on now?

5. What is your role on the team?

6. How long have you worked with team members who are not located at your site?

7. How many team members do you communicate with on a regular basis?

8. How many of those team members are not located at your site?

9. Do you have team members who reside outside the United States?

10. How many team members do you communicate with regularly who reside outside the United States?

11. Would you say your team is using a waterfall or agile software development methodology?

Sources of Advanced Information Technology:

- How much freedom do you have to modify your development process?

- How well-established is your process (i.e., is it new, old, had many versions)?

- Do you feel you have all of the information you need to work within your development process?

- How are decisions made within the development process that you currently use (i.e., team vote, leader decides, depends on the situation)?

- Describe your leadership structure (i.e., hierarchical, matrix).

- Do you feel that your process is efficient? Why or why not?

- How do you resolve conflicts on your project?

- How would you describe the atmosphere of your project?

Other Sources of Structure:

- How are tasks normally executed (i.e., team decides on important tasks and works on them as a group, project manager assigns tasks and team members disperse and work on their assignments)?

- What is the organizational environment (i.e., positive, negative, fast-paced, collaborative)?

Emergent Sources of Structure:

- Do you see any changes in the products you create based on the software development process you are using?

- Has your software development process changed the way you communicate in virtual teams?

- Has the frequency of your communication with virtual team members changed?

New Social Structures:

- How are rules created and modified?

- How are resources allocated? This can include people and technology such as test machines or project repositories.

Social Interaction:

- Do you feel that you use your software development process in its entirety or do you, for example, just use the parts that are the most useful to your team?

- How faithful do you think your team is toward the software development process they are using?

- How does the team use the tools available to them to facilitate the software development process (i.e., wikis, Lotus Notes, Rational Team Concert)?

- How are these tools selected by your team?

- Are there any other tools you would like to use, but are not currently using and why?

- What are the team's attitudes toward these tools?

- How are ideas generated on the team?

- How would you describe the participation of team members in all locations?

- Do you feel that conflict is high, average, or low compared to other teams in your company?

- How do team members influence the project and other members of the team?

- How are tasks typically managed (i.e., status collection, review meetings)?

Group's Internal System:

- What styles of interaction are typically used by the team (i.e., constructive, aggressive, passive)?

- How much knowledge and experience does the team have with the software development process you are using?

- Do team members have sufficient knowledge and experience to do what is asked?

- How much agreement is there between team members on the process used?

Decision Outcomes:

- Do you feel your team has the ability to make efficient decisions?

- Do you think quality decisions are made by your team?

- Do you think consensus is typically achieved in your project? Why or why not?

- Do you feel there is commitment from the team on decisions that are made? Why or why not?

APPENDIX B. CODER TRAINING MATERIALS

All coders were given the interview questions in Appendix A and the information

found in Table 9 below for reference during the coding.

Table 9: AST nodes for coding

Node	Node Color	Discussion Focus
Advanced Information Technology Outputs	Purple	Do you see any changes in the products you create based on the software development process you are using?
Agreement on Appropriation	Pink	How much agreement is there between team members on the process used?
Appropriation Moves	Red	Do you feel that you use your software development process in its entirety or do you, for example, just use the parts that are the most useful to your team?
Atmosphere	Blue	How would you describe the atmosphere of your project?
Commitment	Yellow	Do you feel there is commitment from the team on decisions that are made? Why or why not?
Comprehensiveness	Blue	Do you feel you have all of the information you need to work within your development process?
Conflict Management	Blue	How do you resolve conflicts on your project?
Conflict Mgmt	Red	Do you feel that conflict is high, average, or low compared to other projects?
Consensus	Yellow	Do you think consensus is typically achieved in your project? Why or why not?
Decision Process	Blue	How are decisions made within the development process that you currently use?
Efficiency	Blue	Do you feel your process is efficient? Why or why not?
Efficiency 2	Yellow	Do you feel the team has the ability to make efficient decisions?
Faithfulness of Appropriation	Red	How faithful do you think your team is toward the software development process they are using?

Table 9: Continued

Idea Generation	Red	How are ideas generated by the team?
Influence Behavior	Red	How do team members influence the project and other members of the team?
Instrumental Uses	Red	How are tools selected by your team?
Knowledge and Experience with Structures	Pink	How much knowledge and experience does the team have with the software development process you are using?
Leadership	Blue	Describe your leadership structure.
Level of Sophistication	Blue	How well-established is your process?
Organization Environment	Green	What is the organizational environment? Examples could be positive, negative, collaborative, fast-paced, or something similar that describes how the team works.
Organization Environment Outputs	Purple	Has the frequency of your communication with your virtual team members changed?
Participation	Red	How would you describe the participation of team members in all locations?
Perception of Others' Knowledge	Pink	Do team members have sufficient knowledge and experience to do what is asked?
Persistent Attitudes Toward Appropriation	Red	Are there any tools you would like to use, but are not using and why?
Quality	Yellow	Do you think quality decisions are made by the team?
Resources	Orange	How are resources allocated? Evidence includes both people and technology such as test machines or project repositories.
Restrictiveness	Blue	How much freedom do you have to modify your development process?
Rules	Orange	How are rules created and modified?
Styles of Interacting	Pink	What styles of interaction are typically used by the team? Examples could be constructive, aggressive, or passive.
Task	Green	How are tasks normally executed? For example, does the team decide on important tasks and work on them as a group? Does the project manager assign tasks and the team members disperse and work on their assignments?
Task Management	Red	How are tasks typically managed? Examples could be discussion about status collection or review meetings.
Task Outputs	Purple	Has your software development process changed the way you communicate in virtual teams?

REFERENCES

Abran, A., Moore, J. W., Bourque, P., & Dupuis, R. (Eds.). (2004). *Guide to the Software Engineering Body of Knowledge - 2004 Version* (2004 ed.). Los Alamitos, CA: IEEE Computer Society.

Anderson, L., Alleman, G. B., Beck, K., Blotner, V., Cunningham, W., Poppendieck, M., & Wirfs-Brock, R. (2003). *Agile management - an oxymoron?: Who needs managers anyway?* Paper presented at the Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, Anaheim, CA, USA.

Ashmore, S., & Townsend, A. (2011). *Verbosity and Perceptions of Trust and Leadership in a Chat-based Virtual Team Context.* Paper presented at the DSI 2011, Boston, MA.

Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005). Agile project management: steering from the edges. *Commun. ACM, 48*(12), 85-89.

Baker, G. (2002). The effects of synchronous collaborative technologies on decision making: A study of virtual teams. *Information Resources Management*.

Beck, K. (2000). *Extreme programming explained: Embrace change*. Boston: Addison-Wesley

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Manifesto for Agile Software Development, 2011, from http://agilemanifesto.org/

Bellotti, V., & Bly, S. (1996). *Walking away from the desktop computer: distributed collaboration and mobility in a product design team*. Paper presented at the Proceedings of the 1996 ACM conference on Computer supported cooperative work, Boston, Massachusetts, United States.

Benington, H. D. (1956). *Symposium on advanced programming methods for digital computers.* Paper presented at the Symposium on advanced programming methods for digital computers, Washington D.C.

Bittner, K. (2004). Driving Iterative Development With Use Cases. *IBM developerWorks* Retrieved May 1, 2011, 2011

Bryant, C., & Jary, D. (Eds.). (1991). *Giddens' theory of structuration: A critical appreciation*. London: Routledge.

Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems, 18*(4), 332-343.

Carroll, J. M., & Rosson, M. B. (2002). Usability Specifications as Tool in Iterative Development.

Cascio, W. F. (2000). Managing a Virtual Workplace. *The Academy of Management Executive (1993-2005), 14*(3), 81-90.

Chidambaram, L., & Bostrom, R. P. (1993). Evolution of Group Performance Over Time: A Repeated Measures Study of GDSS Effects. *Journal of Organizational Computing, 3*(4).

Chidambaram, L., Bostrom, R. P., & Wynne, B. E. (1990-1991). A longitudinal study of the impact of group decision support systems on group development. *Journal of Management Information Systems, 7*(3), 7-25.

Cockburn, A. (2002). *Agile software development*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc. .

Cockburn, A. (Producer). (2004). What the agile toolbox contains. *Crosstalk*. Retrieved from http://www.crosstalkonline.org/storage/issue-archives/2004/200411/200411-Cockburn.pdf

Cockburn, A., & Highsmith, J. (2001). Agile software development: the business of innovation. *Computer, 34*(9), 120-127.

Cohen, I. (1989). *Structuration Theory: Anthony Giddens and the constitution of social life*. Houndsmills: Macmillan.

Coplien, J. O., & Harrison, N. B. (2005). *Organizational patterns of agile software development*. Upper Saddle River, NJ: Pearson Prentice Hall

Coppola, N. W., Hiltz, S. R., & Rotter, N. G. (2004). Building trust in virtual teams. *Ieee Transactions On Professional Communication, 47*(2), 95-104.

Craig, L. (2003). Iterative and Incremental Development: A Brief History*, 36,* 47-56.

Damian, D. E., & Zowghi, D. (2003). Challenges in multi-site software development organizations. *Requirements Engineering, 8*(3), 149-160.

DeSanctis, G., & Poole, M. S. (1990). Understanding the use of group decision support systems: the theory of adaptive structuration. In C. S. J. Fulk (Ed.), *Organizations and Communication Technology* (pp. 173-193). Newbury Park, CA: Sage.

DeSanctis, G., & Poole, M. S. (1994). Capturing the complexity in advanced technology use: Adaptive structuration theory. *Organization Science, 5*(2), 121-147.

DiMicco, J., Millen, D. R., Geyer, W., Dugan, C., Brownholtz, B., & Muller, M. (2008). *Motivations for social networking at work*. Paper presented at the Proceedings of the 2008 ACM conference on Computer supported cooperative work, San Diego, CA, USA.

Doshi, C., & Doshi, D. (2009). *A peek into an agile infected culture.* Paper presented at the Agile '09, Chicago, IL.

Drew Procaccino, J., Verner, J. M., Overmyer, S. P., & Darter, M. E. (2002). Case study: factors for early prediction of software development success. *Information and Software Technology, 44*(1), 53-62.

Dul, J., & Hak, T. (2008). *Case study methodology in business research*. Oxford, UK: Butterworth-Heinemann.

Ellram, L. M., Tate, W. L., & Billington, C. (2008). Offshore outsourcing of professional services: A transaction cost economics perspective. *Journal of Operations Management, 26*(2), 148-163.

Eveland, J. D., & Bikson, T. K. (1987). Evolving electronic communication networks: An empirical assessment. *Information Technology & People, 3*(2), 103 - 128.

Ferreira, J., Noble, J., & Biddle, R. (2007). Up-Front interaction design in agile development: Agile processes in software engineering and extreme programming. In G. Concas, E. Damiani, M. Scotto & G. Succi (Eds.), (Vol. 4536, pp. 9-16): Springer Berlin / Heidelberg.

Fiol, C. M., & O'Connor, E. J. (2005). Identification in face-to-face, hybrid, and pure virtual teams: Untangling the contradictions. *Organization Science, 16*(1).

Fowler, F. J. (1995). *Improving survey questions* (Vol. 38). Thousand Oaks: Sage.

Fraser, S., Martin, A., Biddle, R., Hussman, D., Miller, G., Poppendieck, M., . . . Striebeck, M. (2004). *The role of the customer in software development: the XP customer - fad or fashion?* Paper presented at the Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications, Vancouver, BC, CANADA.

Galegher, J., & Kraut, R. E. (1994). Computer-Mediated Communication for Intellectual Teamwork: An Experiment in Group Writing. *Information Systems Research, 5*(2), 110-138.

Gartner. (2006). PPM market universe: Techniques and tools for project collaboration.

Giddens, A. (1977). *Studies in social and political theory*. New York: Basic Books.

Giddens, A. (1986). *The Constitution of Society: Outline of the Theory of Structuration*: University of California Press.

Gruner, K. E., & Homburg, C. (2000). Does Customer Interaction Enhance New Product Success? *Journal of Business Research, 49*(1), 1-14.

Hanssen, G. K., & Erlend F., T. (2006). *Agile customer engagement: a longitudinal qualitative case study*. Paper presented at the Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, Rio de Janeiro, Brazil.

Harter, D. E., Krishnan, M. S., & Slaughter, S. A. (2000). Effects of process maturity on quality, cycle time, and effort in software product development. *Management Science, 46*(4), 451-466

Henninger, S. (1997). Case-Based Knowledge Management Tools for Software Development. *Automated Software Engineering, 4*(3), 319-340.

Hertel, G., Geister, S., & Konradt, U. (2005). Managing virtual teams: A review of current empirical research. *Human Resource Management Review, 15*(1), 69-95.

Hoda, R., Noble, J., & Marshall, S. (2010). *Organizing <i>self-organizing</i> teams*. Paper presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1, Cape Town, South Africa.

Hunt, J. (2006). *Agile software construction*. London, UK: Springer.

Huysman, M., Steinfield, C., Jang, C.-Y., David, K., in 't Veld, M. H., Poot, J., & Mulder, I. (2003). Virtual Teams and the Appropriation of Communication Technology: Exploring the Concept of Media Stickiness. *Computer Supported Cooperative Work (CSCW), 12*(4), 411-436.

Iacono, C. S., & Weisband, S. (1997). *Developing Trust in Virtual Teams.* Paper presented at the Thirtieth Hawaii International Conference on System Sciences, Wailea, HI.

Institute, P. M. (2011). The world is quickly becoming agile. Are you? Retrieved February 20, 2012, 2012, from http://www.pmi.org/Certification/New-PMI-Agile-Certification.aspx

Jacko, J., Düchting, M., Zimmermann, D., & Nebe, K. (2007). Incorporating User Centered Requirement Engineering into Agile Software Development *Human-Computer Interaction. Interaction Design and Usability* (Vol. 4550, pp. 58-67): Springer Berlin / Heidelberg.

Jarvenpaa, S., Knoll, K., & Leidner, D. (1998). Is anybody out there?: antecedents of trust in global virtual teams. *J. Manage. Inf. Syst., 14*(4), 29-64.

Jarvenpaa, S., & Leidner, D. (1999). Communication and trust in global virtual teams. *Organization Science, 10*(6), 791-815.

Jarvenpaa, S., & Leidner, D. E. (1998). Communication and Trust in Global Virtual Teams. *Journal of Computer-Mediated Communication, 3*(4), 0-0.

Johnson, P., Heimann, V., & O'Neill, K. (2001). The "wonderland" of virtual teams. *Journal of Workplace Learning, 13*(1), 24-30.

Kaboli, A., Tabari, M., & Kaboli, E. (2006, August 8–12, 2006). *Leadership in virtual teams*. Paper presented at the The Sixth International Symposium on Operations Research and Its Applications (ISORA'06), Xinjiang, China.

Kahkonen, T. (2004). *Agile methods for large organizations - building communities of practice.* Paper presented at the Agile Development Conference.

Kayworth, T., & Leidner, D. (2000). The global virtual manager: A prescription for success. *European Management Journal, 18*(2), 183-194.

Kayworth, T., & Leidner, D. E. (2002). Leadership Effectiveness in Global Virtual Teams. *J. Manage. Inf. Syst., 18*(3), 7-40.

Kerber, K. W., & Buono, A. F. (2004). Leadership challenges in global virtual teams: Lessons from the field. *SAM Advanced Management Journal, 69*.

Kirkman, B. L., Rosen, B., Gibson, C. B., Tesluk , P. E., & McPherson, S. O. (2002). Five Challenges to Virtual Team Success: Lessons from Sabre, Inc. *The Academy of Management Executive (1993-2005), 16*(3), 67-79.

Klein, H., & Canditt, S. (2008). *Using opinion polls to help measure business impact in agile development*. Paper presented at the Proceedings of the 1st international workshop on Business impact of process improvements, Leipzig, Germany.

Krippendorff, K. (2003). *Content Analysis: An Introduction to Its Methodology*: Sage Publications, Inc.

Larman, C. (2004). *Agile and iterative development: A manager's guide*. Boston, MA: Pearson Education.

Larman, C., & Basili, V. R. (2003). Iterative and Incremental Development: A Brief History. *Computer, 36*(6), 47-56.

Law, A., & Ho, A. (2004). *A Study Case: Evolution of Co-Location and Planning Strategy*. Paper presented at the Proceedings of the Agile Development Conference.

Lehman, M. M., & Belady, L. A. (1985). Internal IBM Report, 1969 *Program evolution: processes of software change*. San Diego, CA: Academic Press Professional, Inc.

Lind, M. R. (1999). The gender impact of temporary virtual work groups. *Professional Communication IEEE Transactions, 42*(4), 276-285.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., . . . Kahkonen, T. (2004). Agile software development in large organizations *Computer, 37*(12), 26 - 34.

Lipnack, J., & Stamps, J. (1997). *Virtual Teams-Reaching Across Space, Time, and Organizations with Technology*: 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.

Majchrzak, A., Rice, R. E., Malhotra, A., King, N., & Ba, S. (2000). Technology Adaptation: The Case of a Computer-Supported Inter-Organizational Virtual Team. *MIS Quarterly, 24*(4).

Martins, L., Gilson, L., & Maynard, M. T. (2004). Virtual Teams: What do we know and where do we go from here? *Journal of Management, 30*(6), 805-835.

Maznevski, M. L., & Chudoba, K. M. (2000). Bridging Space Over Time: Global Virtual Team Dynamics and Effectiveness. *Organization Science, 11*(5), 473-492.

McConnell, S. (2004). *Code Complete: A Practical Handbook of Software Construction*. Redmond, WA: Microsoft Press.

McDonough, E. F., Kahnb, K. B., & Barczaka, G. (2001). An investigation of the use of global, virtual, and colocated new product development teams. *Journal of Product Innovation Management, 18*(2), 110-120.

Miles, M., & Huberman, A. M. (1984). *Qualitative data analysis: A sourcebook of new methods*. Beverly Hills: Sage

Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869-1890.

Mnkandla, E., & Dwolatzky, B. (2006). *Defining agile software quality assurance.* Paper presented at the International Conference on Software Engineering Advances, Tahiti.

Montoya-Weiss, M. M., Massey, A. P., & Song, M. (2001). Getting it together: Temporal coordination and conflict management in global virtual teams. *The Academy of Management Journal, 44*(6), 1251-1262

Murphy, T. E., Duggan, J., Norton, D., Prentice, B., Plummer, D. C., & Landry, S. (2009). Predicts 2010: Agile and cloud impact application development directions In G. Research (Ed.). Stamford, CT.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Commun. ACM, 48*(5), 72-78.

Nguyen, T. (2006). A decision model for managing software development projects. *Information & Management, 43*(1), 63-75.

Ollman, B. (1971). *Alienation: Marx's Conception of Man in Capitalist Society*. Cambridge: Cambridge University Press.

Poltrock, S., & Grudin, J. (1998). *CSCW, groupware and workflow: experiences, state of art, and future trends*. Paper presented at the CHI 98 conference summary on Human factors in computing systems, Los Angeles, California, United States.

Powell, A., Piccoli, G., & Ives, B. (2004). Virtual teams: a review of current literature and directions for future research. *SIGMIS Database, 35*(1), 6-36.

Prikladnicki, R., Audy, N., Luis, J., & Evaristo, R. (2003). Global software development in practice lessons learned. *Software Process: Improvement and Practice, 8*(4), 267-281.

Raccoon, L. B. S. (1997). Fifty years of progress in software engineering. *SIGSOFT Software Engineering Notes, 22*(1), 88-104.

Raghuram, S., Tuertscher, P., & Garud, R. (2010). Research Note---Mapping the Field of Virtual Work: A Cocitation Analysis. *Info. Sys. Research, 21*(4), 983-999.

Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006). Can distributed software development be agile? *Commun. ACM, 49*(10), 41-46.

Riffe, D., Lacy, S., & Fico, F. (2005). *Analyzing media messages: using quantitative content analysis in research* (2 ed.). New Jersey: Lawrence Earlbaum Associates.

Rus, I., & Lindvall, M. (2002). Guest Editors' Introduction: Knowledge Management in Software Engineering, *19,* 26-38.

Sanders, E. (2002). From user-centered to participatory design approaches. In J. Frascara (Ed.), *Design and the social sciences: making connections*. New York, NY: Taylor and Francis Inc.

Sarker, S., Lau, F., & Sahay, S. (2001). Using an Adapted Grounded Theory Approach for Inductive Theory Building About Virtual Team Development. *Database for Advances in Information Systems, 32*(1), 38-56.

Sawyer, S. (2004). Software development teams. *Communication of the ACM, 47*(12), 95-99.

Schmidt, K., & Bannon, L. (1992). Taking CSCW seriously. *Computer Supported Cooperative Work (CSCW), 1*(1), 7-40.

Scott, M., & DeSanctis, P. G. (1992). Microlevel Structuration in Computer-Supported Group Decision Making. *Human Communication Research, 19*(1), 5-49.

Shaw, M. (2000). *Software engineering education: a roadmap*. Paper presented at the Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland.

Snow, C. C., Snell, S. A., Davison, S. C., & Hambrick, D. C. (1996). Use transnational teams to globalize your company. *Organizational Dynamics, 24*(4), 50-67.

Stones, R. (2005). *Structuration theory*. Houndmills: Palgrave Macmillan.

Succi, G., & Marchesi, M. (Eds.). (2001). *Extreme programming examined*. Boston: Addison-Wesley Longman Publishing Co., Inc.

Suchan, J., & Hayzak, G. (2001). The Communication Characteristics of Virtual Teams: A Case Study. *IEEE Transactions on Professional Communication Research, 44*(3).

Tan, B. C. Y., Wei, K.-K., Huang, W. W., & Ng, G.-N. (2000). A dialogue technique to enhance electronic communication in virtual teams. *Professional Communication IEEE Transactions, 43*(2), 153-165.

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.

Townsend, A. M., Hendrickson, A. R., & DeMarie, S. M. (2002). Meeting the virtual work imperative. *Commun. ACM, 45*(1), 23-26.

Tullar, W. L., & Kaiser, P. R. (2000). The Effect of Process Training on Process and Outcomes in Virtual Groups. *Journal of Business Communication, 37*(4), 408-426.

Van Ryssen, S., & Godar, S. H. (2000). Going international without going international: multinational virtual teams. *Journal of International Management, 6*(1), 49-60.

Wagstrom, P., Martino, J., Kaenel, J. v., Chetty, M., Thomas, J., & Jones, L. (2011). *A dive into online community properties*. Paper presented at the Proceedings of the ACM 2011 conference on Computer supported cooperative work, Hangzhou, China.

Walther, J. (1995). Relational aspects of computer-mediated communication: Experimental observations. *Organization Science, 6*, 186-203.

Walther, J. B., & Tidwell, L. C. (1995). Nonverbal cues in computer-mediated communication, and the effect of chronemics on relational communication. *Journal of Organizational Computing, 5*(4), 355 - 378.

Warkentin, M., & Beranek, P. M. (1999). Training to improve virtual team communication. *Information Systems Journal, 9*(4), 271-289.

Westerlund, M., Normark, M., & Holmquist, L. E. (2011). *Express location: supporting coordination of mobile delivery work*. Paper presented at the Proceedings of the ACM 2011 conference on Computer supported cooperative work, Hangzhou, China.

Whiteside, J., Bennett, J., & Holzblatt, K. (1988). Usability engineering: Our experience and evolution. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 791-817). Amsterdam, North Holland.

Williams, L., & Cockburn, A. (2003). Guest Editors' Introduction: Agile Software Development: It's about Feedback and Change. *Computer, 36*(6), 39-43.

Woodward, E., Surdeck, S., & Ganis, M. (2010). *A practical guide to distributed scrum*. Boston, MA: Pearson Education, Inc.

Yin, R. K. (2009). *Case study research: Design and methods* (Vol. 5). Los Angeles, CA: Sage.

Yoo, Y., & Alavi, M. (2004). Emergent leadership in virtual teams: what do emergent leaders do? *Information and Organization, 14*(1), 27-58.

Zurcher, F. W., & Randell, B. (1968). *Iterative multi-level modeling: A metholodology for computer system design.* Paper presented at the Proceedings IFIP Congress 1968.