An exploratory examination of modes of interaction and work in waterfall and agile teams

Sondra Ashmore*

W.R. Berkley Corporation,
3840 109th Street,
Urbandale, Iowa 50322, USA
Email: sondra1130@yahoo.com
*Corresponding author

Anthony Townsend, Samuel DeMarie and Brian Mennecke

Iowa State University Ivy College of Business,
Gerdin Business Building, 1200,
2167 Union Drive, Ames, Iowa 50011, USA
Email: amt@iastate.edu
Email: demarie@iastate.edu
Email: mennecke@gmail.com

Abstract: As agile becomes the preferred methodology for managing projects, many organisations are trying to understand how their processes and culture will change. Using the adaptive structuration theory as a lens for analysis, this case study explores the dynamics of two large teams, one who continued to use the Waterfall method and other who transitioned to agile methodologies. The results show that there are important adaptations and cultural differences that should be considered when an organisation starts leveraging agile methods.

Keywords: agile; waterfall; software development; adaptive structuration theory; organisational change; virtual teams.

Reference to this paper should be made as follows: Ashmore, S., Townsend, A., DeMarie, S. and Mennecke, B. (2018) 'An exploratory examination of modes of interaction and work in waterfall and agile teams', *Int. J. Agile Systems and Management*, Vol. 11, No. 1, pp.67–102.

Biographical notes: Sondra Ashmore is an experienced IT leader with expertise in IT strategy, UX and software development methodologies. She received her Master's degree at the Rensselaer Polytechnic Institute in Technical Communication and Management and PhD at the Iowa State in Human Computer Interaction. Her research focuses on the software development process, both waterfall and agile and explores strategies to optimise the user experience. She was recognised as a 'Forty under 40' Business Leader by the Business Record and won the Iowa Technology Association's Women of Innovation Award for business innovation and leadership. She co-authored *Introduction to Agile Methods*.

Anthony Townsend is an Associate Professor in Information Systems and Human-Computer Interaction at Iowa State University. His research focuses on virtual teams, software development processes and ethics. He received his PhD in Organisational Behaviour and Industrial Relations from Virginia Tech.

Samuel DeMarie is a leading expert in business strategy, culture, leadership and virtual teamwork. His research is widely cited and published in many leading academic journals and has been presented at numerous international conferences. He has worked with a wide range of organisations in the areas of designing and executing more effective strategies and building stronger company cultures. He provides professional consulting services through DeMarie Management group LLC. He is currently an Associate Professor of Management in the College of Business at Iowa State University. He teaches business strategy courses in both the MBA and undergraduate programs. He earned his PhD in Strategic Management from the Arizona State University.

Brian Mennecke was a Professor of Business and Information Systems at the Iowa State University. His research focused on usability, online communication and virtual environments. He received his PhD in Information Systems from the Indiana University.

1 Introduction

As firms grapple with the increased dynamism present in their competitive environments, new modes of organising the work of teams and other organisational units have been devised and applied to identify problems, design solutions and implement change. Over the last decade, the agile methodology has gained prominence as an alternative approach to software development. For example, research by the Project Management Institute (PMI) suggests that the use of agile methodologies has tripled in the last several years (PMI, 2014), which parallels research by other industry analysts suggesting that in the next few years, the vast majority of software development projects will use agile methodologies (Murphy et al., 2009; Wilson, 2013).

While much of the literature has examined how agile should be deployed, little research has focused on the impact of agile methodologies on the team's culture and working structures (e.g., Chow and Cao, 2008; Conboy et al., 2011; Procaccino et al., 2002; Fitzgerald et al., 2006; Hosalkar and Bowonder, 2000; Meso and Jain, 2006; Misra et al., 2009; Nielsen and McMunn, 2005; Nerur et al., 2005; Rohunen et al., 2010; Sarker and Sarker, 2009; Sidky et al., 2007; Srinivasan and Lundqvist, 2009; Tan and Teo, 2007). For example, while research has examined issues such as why an organisation might adopt agile methodologies (Lee, 2008; Misra et al., 2006), the impact of that choice on teams' cultures and structures remains unexplored. This leads us to our two research questions:

Research question 1	How does each methodology, agile and waterfall, influence the structural context of software development groups?
Research question 2	How does appropriating a particular methodology affect culture among members of software development groups?

To examine these questions, we studied two large development work groups (DWG)[1] within one firm. Both DWGs traditionally used the waterfall methodology; however, while one of the groups continued their use of waterfall, the second group had adopted agile methodologies. Because both DWGs are from the same firm and both have members with significant experience using the waterfall methodology, our examination of the differing impacts of the two methodologies on group structure and culture represents a unique opportunity to understand how organisational work and attitudes are influenced by the use of each of these methodologies.

To examine these issues, we frame our data collection and analysis using adaptive structuration theory (AST) (DeSanctis and Poole, 1994), which offers both a framework for the collection of our data and also a lens through which the data can be interpreted. AST has been widely used in teams research (Raghuram et al., 2010) as well as software development studies that involve teams (Cao et al., 2009; Majchrzak et al., 2000). We specifically build on the work of Cao et al. (2009), who used AST as a framework for investigating how software development teams adapt to agile methodologies. We do by examining two different sets of software development groups from the same organisation that used two different methodologies (agile and waterfall), which allows us to compare how these methodologies influence team structure and culture. Our results offer important insights about how methodologies influence the organisation of work structures and culture in teams, which has important implications for team leaders seeking to lead teams using waterfall, agile, or other hybrid methodologies.

2 Background

We begin our discussion of the relevant literature by reviewing the pertinent research and practices associated with software development. We follow this with a discussion of adaptive structuration theory, which is the theoretical lens used to frame our examination in this research.

2.1 Software development methodologies

One of the most popular software development processes is the waterfall methodology. The waterfall methodology advocates an approach that has a focus on structuring the work process using sequential phases of development with each stage completing before the next begins. For example, all software designs are completed before the coding phase begins and all coding is completed before testing is started. The waterfall methodology was first introduced in the 1950s for the military and it continues to be one of the more commonly used approaches to software development (Benington, 1956; Larman and Basili, 2003). Despite the popularity of waterfall development, it has continued to be criticised for being process heavy and unresponsive to the inevitable changes that arise during software development projects (McConnell, 2004).

As e-commerce matured, corporate sites increased in sophistication and smart phone applications proliferated, product development teams needed new ways to quickly respond to these demands to remain competitive in changing markets. One prominent solution came in the form of agile development, validating previous assertions by

Whiteside and Bennett that software development needed to become more of an iterative process (Whiteside et al., 1988). In fact, agile development is commonly referred to as iterative development and is the overarching term for processes such as agile, lean, extreme programming, scrum and rational unified process (Larman, 2004). Each of these elements of agile development are unique, but they share the common goal of avoiding "a single pass sequential, document-driven, gated-step approach" [Larman and Basili, (2003), p.47]. Most of the time the term agile is used to describe iterative approaches to software development that embrace the values of the Agile Manifesto. 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.

Table 1 Traditional (waterfall) versus agile

	Traditional	*Agile*
Fundamental assumptions	Systems are not 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 and 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-favours specialisation	Self-organising 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 Organisational	Mechanistic (bureaucratic with high formalisation)	Organic (flexible and participative encouraging cooperative social action)
Technology	No restrictions	Favours object-oriented technology

Source: Adapted from Nerur et al. (2005)

This issue of culture is important because it will be related not only to perceptions and attitudes, but it is also likely to be reflected in the team's work behaviours and actions. For example, an important difference between agile development and waterfall development is how project phases are scheduled (see Table 1). In the waterfall methodology, it is common for a project phase to take several months to brainstorm,

design, modify, review and, finally, approve. It is important to note that when approvals are secured, the design for the entire project is largely locked and a formal change request must be made to change the design after approval. This process implies that waterfall generally leads to a more rigid, formal and routine set of activities, which will influence the culture of the team as well as how the development group approaches and implements solutions to the problem.

In contrast, a typical agile design phase is much shorter (i.e., a matter of a week or two) and 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 describe the time a project team is allotted to complete a particular portion of a product from start to finish. The iterative development process embedded in the agile methodology encourages customers to be influential with the end product during most or all phases of the development lifecycle rather than just at the beginning (Bittner, 2004; Düchting et al., 2007). The implication is that agile's focus on shorter timeframes and responsiveness to stakeholders will lead to a culture and work behaviours that are more open, collaborative and flexible than would be the case for a development group using waterfall. For example, team members that are using agile typically meet daily in 'scrums' to discuss priorities, goals and customer feedback (Woodward et al., 2010). This cadence differs significantly from the waterfall approach, where group members tend to hold status meetings weekly or, sometimes, less frequently.

2.2 Adaptive structuration theory

Adaptive structuration theory (AST) is derivative of structuration theory (Giddens, 1977), which was originally proposed to facilitate the investigation of how and why societal structures remain in some cases and dissolve in others. Structuration theory differs from sociological theories that preceded it in that it supposes that human behaviour is not based on the actions of individuals or society as a whole, but rather the norms that are developed within unique structures. These structures and the rules that govern them are malleable and can be expected to change over time (Giddens, 1986).

AST was initially introduced as a framework for studying the interaction of groups and organisations with information technology (DeSanctis and Poole, 1990). AST offers that a group or organisation's perceived utility in combination with the benefit of a technology drive its outcomes and future use. This perception also influences how perceptions and use of the technology changes over time as the group or organisation appropriates features of the technology. The key components comprising AST are Giddens' (1977) model of structuration as well as Ollman's (1971) concept of appropriation. Together, these concepts illustrate how the groups or organisations adapt the artefact to meet their collective and individual goals and requirements. The model shown in Figure 1 represents the seven propositions that form expectations about how individuals adapt to an advanced technology artefact.

AST has been used as a theoretical framework in both 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). Importantly, Cao et al. (2009) applied AST to investigate how software development teams adapt to agile methodologies. They suggested that four sources of structure influence the appropriation of agile development methodology:

1 agile methods defined through their structural features and spirit

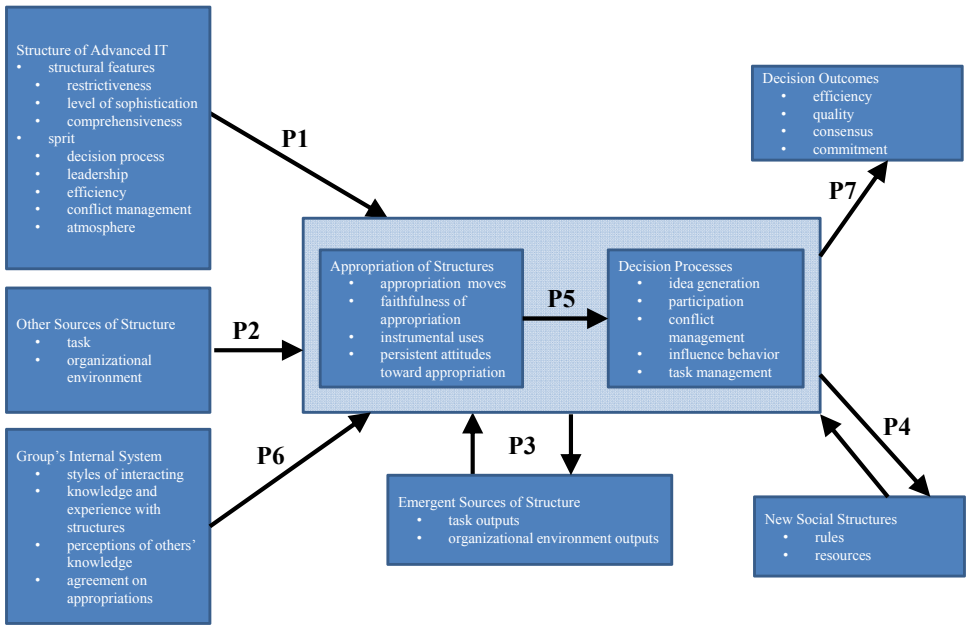
2 software project characteristics

3 organisational context

4 each team's internal system that includes their interaction style, knowledge, expertise with and perceptions about agile methods.

In their study, they analysed interview transcriptions and code based on the Adaptive Structuration model and found four major appropriation practices: development process related, developer related, customer related and organisation/management related (Cao et al., 2009).

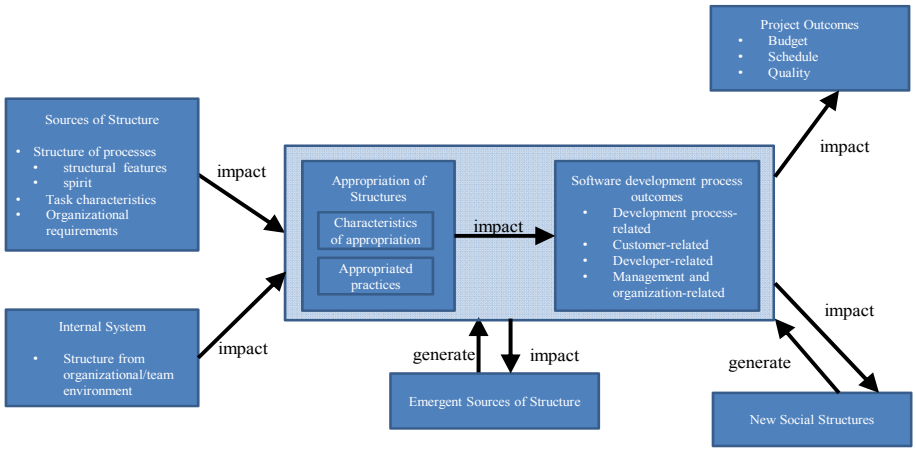
Figure 1 AST constructs and propositions (see online version for colours)



Source: Adapted from DeSanctis and Poole (1994)

Cao et al.'s (2009) study focused on the software development process (agile) as the focal artefact that was adopted and appropriated by their agile development teams. Their paper offers a justification for extending the use of AST to an examination of appropriations of and adaptions to new product development methodologies. Our research is motivated in part by Cao and colleagues' application of AST and we apply their adapted model of AST (see Figure 2) in a new context by comparing the appropriation of work patterns and cultural perspectives made by development groups that chose to use differing development methodologies (i.e., waterfall versus agile). In other words, by examining two different sets of groups using two different methodologies, we can examine how these methodologies influence structures and culture.

Figure 2 AST model adapted for agile development (see online version for colours)



Source: Adapted from Cao et al. (2009)

3 Propositions

Previous research examining teams, software development processes and AST provide a theoretical basis for predicting the likely differences that will be found between teams using agile development methodologies versus those using waterfall software methodologies. Cao et al. (2009) adapted the AST model by focusing on six propositions and by eliminating 'other sources of structure', which they deemed to be less influential in agile development teams. Specifically, they suggest that the following are the most influential structures influencing teams when they move from waterfall to agile software development:

1 agile methods defined through their structural features and spirit;

2 software project characteristics;

3 organisational context; and

4 each team's internal system, which includes their interaction style, knowledge and expertise with agile methods and their perceptions about agile methods.

Given the relevance of their framework for understanding agile appropriations, we use their framework as a guide in the design of our research and for laying out expectations for our findings. Specifically, we offer two propositions and related hypotheses that are applied from Cao et al. (2009) model to the context of our research. The first proposition addresses how the development group members will apply structures in their appropriation of agile and the second examines how the appropriations of agile influence the each group's culture. Each proposition is now discussed in turn.

Proposition 1 The DWG using the agile methodology will demonstrate key differences in structures, spirit and attitudes from the waterfall DWG as a result of the appropriation of different methodologies.

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). Cao et al. (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. They argue that the structural features and spirit constructs under the 'Sources of Structure' component are among the most important differentiating constructs for agile teams in their agile framework. As applied to a comparison of teams within either the agile or waterfall DWGs, this suggests the following:

Hypothesis 1a The agile DWG will demonstrate structural features and spirit that differ from the waterfall DWG, given the differences in processes and values between the two methodologies.

AST differentiates structural features from the organisational environment if the organisational environment provides the contextual structures in which structural features are housed (DeSanctis and Poole, 1994). Cao et al. (2009) argue that the organisational context of an agile environment is one of decentralised decision-making and flattened organisational structures. Waterfall teams are usually hierarchical in structure and require that changes be approved through a very specific set of stakeholders (Raccoon, 1997). In comparing our agile and waterfall development groups in terms of decision-making processes, this suggests the following:

Hypothesis 1b Members of the agile DWG will report that their structure is less hierarchical and their decision processes are less centralised than the waterfall DWG.

An internal system in the AST model describes the nature of the members and their relationships inside the team (DeSanctis and 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 and Highsmith, 2001). Members of a DWG will have adapted to a new internal system after having moved from a waterfall to an agile methodology.

Hypothesis 1c The agile DWG's internal system will be more collaborative than the waterfall DWG's internal system.

AST defines attitude as "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 and Poole, (1994), p.130]. In this study, we use attitude toward the development process rather than attitude toward a technology. Many members of the DWGs using agile development methodologies 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). Members of waterfall DWGs may engage in review sessions and other retrospectives and may be trained in a particular technique, but the frequent focus on review and self-critique of processes will mean that members of agile DWGs will have more confidence in and positive feelings about the process.

Hypothesis 1d Members of the agile DWG will have a more positive attitude about the agile development methodology than the members of the waterfall DWG will have about their methodology.

Concerning culture and Proposition 2, Doshi and Doshi (2009) found that moving to an agile development methodology not only changes the development process, but also changes the culture of the team. The Agile Manifesto 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 emphasises 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 (2000) also suggest that the cultural composition of a team (i.e., the agile culture) is an influential structural characteristic.

Proposition 2 The appropriations by members of the agile DWG will create a culture that is unique and distinct from those in the waterfall DWG.

The Agile Manifesto that was created and advocated by the early adopters of agile development emphasised 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) emphasise that agile teams focus on individual competencies and increasing collaboration levels. This suggests that members of agile DWGs will be more likely to have positive collaborative experiences and feel better about their interactions.

Hypothesis 2a Members of the agile DWG will describe their atmosphere as more collaborative than will members of the waterfall DWG.

Agile development requires that a team adopt new values and team dynamics (Beck et al., 2001; Doshi and Doshi, 2009) that will shift the social interaction constructs of a team. Virtual team research has found that the richer the media used in the interaction, the more cohesive will be the feelings of team members (Fiol and O'Connor, 2005), the more social presence the team member will feel (Baker, 2002) and the more trust that will develop between team members (Warkentin and Beranek, 1999). This suggests that a global DWG using the agile development methodology will make more appropriations of richer forms of communication media in order to overcome their inability to meet physically. This is consistent with findings from Singh et al. (2015) that found using agile methods encourages more regular communication. For members of a waterfall development group, the need for richer forms of communication will be less pronounced because this methodology does not call for interactions that are as rich or frequent compared to agile; therefore, waterfall group members will be less likely to report the need to appropriate richer forms of communication media.

Hypothesis 2b Members of the agile DWG will have a greater orientation toward appropriating technologies that support collaboration, cohesion and trust as compared to the members of the waterfall DWG.

Cockburn and Highsmith (2001) emphasise 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.

Hypothesis 2c Members of the agile DWG will state that they have more control over the quality and outcome of the product they are creating compared to members of the waterfall DWG.

Members of the agile DWGs, specifically those using the Scrum and pair 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 no 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).

Hypothesis 2d Members of the agile DWG will have more interactions between members than will the members of the waterfall DWG.

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 and Highsmith, 2001; Williams and 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).

Hypothesis 2e The agile DWG will be more likely to embrace project change and make appropriations that support change compared to the waterfall DWG.

Taken together these hypotheses provide the basis of our examination of the impact of development methodologies on culture and structure. Prior research has not controlled for organisational effects on agile team structure, in that they have not examined structures of non-agile teams in the same organisation as the agile teams; our hypotheses are designed to examine that comparison and identify structures that are the result of the adoption of the agile methodology.

3.1 The study site

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, is cutting-edge in global DWG's and has encouraged its DWGs to move from the waterfall development methodology to agile. This company started deploying agile projects in 2005 and they have invested heavily in training select DWGs on agile best practices. At the time that we collected these data, agile development was not required for all projects but it was strongly encouraged. Companywide, there was a good mix of teams and development groups that either were exclusively using the waterfall methodology, using a hybrid of waterfall and agile methodologies, or had converted completely to an agile approach.

The decision to use the waterfall methodology or to move to the agile methodology was ultimately decided collectively by the team members within each DWG. Executive

management encouraged and enabled the adoption of agile, but entrusted team members in each DWG to select the process that was best suited for each project. This company deployed agile coaches to help their development groups' transition seamlessly to agile. These coaches worked with a wide variety of 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.

3.2 Participants

Two large DWGs were the focus for this study. Each DWG worked on their own distinct product and each group includes several global work teams. Each DWG reports to the same director, but they have different first and second-level managers. Members did not self-select the project DWG on which they worked, but most had remained on the same DWG for many project cycles. Both DWGs are working on software products that have been on the market a number of years and have an established customer base. Both DWGs have a long history of using waterfall to develop software and are working on projects of similar complexity; however, the defining difference between the two DWGs is that one DWG (the agile DWG) had moved to the agile development methodology prior to the beginning of the current project. Similarly, the other group (the waterfall DWG) had made the decision to retain the waterfall methodology prior to beginning the current project. These two large globally-distributed DWGs were ideal for examining differences associated with the two software development approaches and AST is an ideal lens through which to examine and interpret our observations. A description of each group follows and a summary of the characteristics of the two DWGs is provided in Table 2.

Table 2 Characteristics of DWGs

Characteristic	*Agile DWG*	*Waterfall DWG*
Time in current method	Nine months on current project, greater than 12 years before on waterfall	Greater than 15 years, nine months on current project
Number of members in DWG	191	268
Number of teams within DWG	12	No formal team divisions, matrix structured
Team size	6–19	N/A

3.3 The waterfall DWG

The waterfall DWG is working on a product that backs up and recovers critical data for organisations. Their product has the goal of backing up data to disk and tape drives to ensure organisations retain their required information for business and compliance purposes. Their product spans a large customer base and is the highest revenue driver for the organisation. The nature of their product and its importance to their organisation's bottom line mean that quality and reliability are critically important.

At the time of the interviews, members of this group were engaged in a large multi-year project. The design of an effective user interface (i.e., a focus on user experience) was an important aspect of this project; however, technical quality and reliability were critical success factors for this release. Project requirements were pre-determined by market requirements and customer feedback from previous projects.

The DWG as a whole consisted of 268 group members worldwide. Group members have worked on previous iterations of the product together using a waterfall methodology and their average experience using waterfall methodology was slightly more than 20 years. Prior to the start of the current project, this group had made the decision to retain the waterfall methodology; as a result, it had not formally subdivided into smaller teams. Instead, it retained a matrix structure consistent with its extant methodology. Members of this group indicated that they had optimised their environment and processes based on lessons learned in previous projects. They communicated primarily using an instant messaging tool, e-mail and teleconferences. Project status was typically reviewed on a weekly basis, with executive reviews on a monthly cadence.

The waterfall DWG's tools of choice are internally developed databases (i.e., repositories) 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 that they had used in previous projects. Members had received extensive training on the waterfall software development methodology and they are regularly audited by the company for process compliance.

Members of this DWG have had significant, corporate-mandated training on the agile development methodology. Although the DWG was fully trained in agile, managers who oversaw this group indicated that the group members themselves preferred that other organisational development groups 'try out' the agile methodology before this group reconsidered transitioning to agile, because of the project's critical importance to meeting overall firm goals. As a result, members of this DWG were focused on using waterfall techniques for the foreseeable future.

3.4 The agile DWG

The agile DWG that we studied was an early adopter of agile within the organisation and was undertaking a large project. The primary agile methodology that the DWG adopted was Scrum (Schwaber and Beedle, 2002). Scrum is one of the most popular and widely adopted agile methodologies (Ashmore and Runyan, 2014) due to the focus it puts on team communication, requirements prioritisation and short development cycles. According to informants, this DWG leveraged the following from Scrum:

- Team dynamics – initially organised in Scrum teams by feature, then adapted to Scrum teams in similar time zones. Each team assigned a Scrum master to keep the team focused. Each Scrum master attended a 'Scrum of Scrums' meeting to facilitate communication among the various Scrum teams working on the project.
- Project requirements management – the DWG committed to themes such as 'improved user experience' rather than specific requirements at the beginning of the project to allow for flexibility based on customer feedback. Specific requirements were written in each Scrum's user story format and prioritised in a product backlog.

- Short development cycles – the teams within the DWG started by using three week development cycles or 'sprints' and then ultimately moved to five week sprints based on team member feedback. Sprints were tracked daily using burn down charts.

The agile DWG also used elements of the extreme programming (XP) agile methodology (Beck, 2000). XP encourages working code rather than extensive documentation, frequent testing, regular builds and learning from teammates by programming in pairs, which is often called pair programming. The agile DWG implemented daily rather than weekly builds and used pair programming to improve code quality and to provide training among group members. As is typical, the DWG used the elements of multiple agile methods to construct their operating methodology (Conboy, 2009).

The agile DWG worked on a product that monitors data centres and provides disaster recovery solutions. The project, the latest release of the software, was expected to be over three years in duration and was not completed by the conclusion of the data collection phase of this research project. The agile DWG consisted of 191 members divided into 12 scrum teams. Members had worked together on previous releases of this product, but they had adopted the agile development methodology for the first time at the start of the current project, more than a year prior to our data collection.

Implementing the agile methodology required extensive changes in the group's meeting and group dynamics, tooling, processes and mindset to make the transition possible. Training on the agile development process was available for all of the 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. In addition, a professional agile coach, who was also employed at the company, was available to help members transition to agile. The members of this group placed a significant focus on documenting best practices. Most members indicated that they wanted to understand how they could improve their processes and appropriations of agile for future projects. They also expressed an interest in sharing their experiences with other DWGs in the organisation that might subsequently move to agile.

The agile DWG and its teams selected and used a software product called Rational Team Concert (RTC), which was the recommended, but not required, project tracking tool at their company. RTC brings together the tools required to deliver a product supportive of the agile methodology. The 'team organisation' feature was used for communication about user stories. This included feedback on user stories, approvals and any changes that needed to be made. This DWG also made extensive use of 'artefacts' in RTC, which allowed them to link the code repositories with project status for real-time tracking and also track code defects. One key requirement was the ability to track earned value by showing the actual story points completed compared to the projected story points completed by sprint iteration. Training on RTC was provided (by the firm) to the entire DWG before the start of the project.

This large DWG was divided into 12 scrum teams (with between six and 19 members each) responsible for different features of the product. All DWG members had worked on software development projects prior to participating on this project. While some Scrum members were co-located, all Scrum teams had globally-distributed members. A 'scrum of scrums' was also held that included all of the Scrum Masters. Each Scrum consists of male and female team members that range in experience from new hires to experienced professionals.

4 Methodology

In this section we present the methodology we used to examine these hypotheses. We first discuss the case study methodology we used to examine the research questions and hypotheses. Next, we discuss the study site, which involved two development groups operating in the same organisation.

4.1 Case study research design

A case study design was chosen for this project because it best fits the nature of the investigation. The case study approach has been used as a tool to better understand how agile methods impact organisations (e.g., Ashmore and Wedlake, 2016; Grimaldi et al., 2016). Case studies are most relevant for research questions that focus on 'how and why', where there is no control over behavioural events and finally where the research focuses on contemporary events (Yin, 1995). This project focused on the overarching questions of *how* do agile and waterfall differ and *why* are these differences important? The project took place in a field setting (with no behavioural control) and involved a very contemporary issue. An embedded single case design was determined to be the best choice to investigate these issues. By looking at two sub-groups embedded in the same company we were able to avoid problems related to differences dependent on organisational setting and culture.

We used DeSanctis and Poole's AST (1990, 1994) as the theoretical framework for the design our data collection activities and to analyse and interpret the data we collected. DeSanctis and Poole's seminal article on AST advocates for the use of written transcript and audio recordings to capture the words of participants for categorisation and interpretation (1994, p.139). The case study approach has been used in similar studies (Cao et al., 2009). Consistent with Cao et al. (2009) use of AST to study software development teams, this study focuses on the software development methodologies' impact on the group's structural and cultural adaptations.

4.2 Data collection

Eleven focus groups (five from the waterfall DWG and six from the agile DWG) were conducted with the waterfall and agile DWG members. For each focus group, all participants were from either an agile DWG or a waterfall DWG, but not both. Smaller focus groups were used to accommodate the schedules of the participants, but ultimately the responses were aggregated for analysis and comparison between the two types of DWGs, agile and waterfall. Additional interviews with second line managers, project executives and the Development Director were conducted to capture leadership perspectives. 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 members located in the US and 44 agreed to participate, yielding a 57% participation rate.

Members of the research team conducted the one-hour interviews. 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. A set of structured interview questions was used to guide the focus group discussions around the research questions. The interview questions can be found in Appendix A.

Focus groups were separated by process type used (waterfall or agile) and then broken down again by similar roles in the organisation. 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. Executives were interviewed separately because they are not an active part of the day-to-day interactions of the DWG's. Further, because many of the executives manage both waterfall and agile WDGs, their responses might be prejudiced based on their association with both types of groups. Additionally, segmenting the focus groups by rank was prompted by concerns 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.

Yin (2009) and Fowler (1995) emphasise the importance of conducting a pilot study to refine the procedures and interview questions. The interview questions were piloted with a representative group of three team members in the software development organisation from the company to check for clarity and appropriateness of the questions. Responses from the pilot sessions were not included in the study's final results. Some of the questions were modified for clarity based on feedback from the pilot, but none of the questions were eliminated. The final list of questions used during interviews is shown in Appendix A.

The interviews were conducted using in-person focus groups held in conference rooms at two of the major development sites over a two-day period. Interviews were not always kept entirely to the scripted questions because follow-up and clarifying questions were often needed to understand the responses or to fill in context. The interview questions were designed to help the researchers understand the business context and group dynamics through the lens of AST. Individual participants 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 (e.g., Scott and DeSanctis, 1992). Project documents were also reviewed to supplement the information that was provided during the interviews.

5 Results

After the focus group interviews were completed, the recordings and notes were reviewed, transcribed and analysed using the process outlined by DeSanctis and Poole (1994). Interview transcriptions were coded using NVivo (http://www.qsrinternational.com/products_nvivo.aspx) by the investigators in accordance with coding guidelines and using a coding book (see Appendix B). The initial coding was subsequently validated by two additional coders. Each coder was given training on AST, the survey questions, the nodes and general training on the NVivo coding tool. Transcripts were categorised as waterfall or agile, depending on the background of the participants in the interview.

Table 3 Summary of results

Hypothesis		*Nodes used in analysis*		*Supported*
1a	The agile DWG will demonstrate structural features and spirit that differ from the waterfall DWG, given the differences in processes and values between the two methodologies.	• Restrictiveness • Level of sophistication • Comprehensiveness	1a	Yes
1b	Members of the agile DWG will report that their structure is less hierarchical and their decision processes are less centralised than the waterfall DWG.	• Decision process • Leadership • Efficiency • Conflict management • Atmosphere	1b	Yes
1c	The agile DWG's internal system will be more collaborative than the waterfall DWG's internal system.	• Styles of interacting • Knowledge and experience with structures • Perception of others' knowledge • Agreement on appropriation		Yes
1d	Members of the agile DWG will have a more positive attitude about the agile development methodology than the members of the waterfall DWG will have about their methodology.	• Persistent attitude toward appropriation • Faithfulness of appropriation		Yes
2a	Members of the agile DWG will describe their atmosphere as more collaborative than will members of the waterfall DWG.	• Atmosphere • Idea generation • Participation • Consensus • Conflict management • Influence behaviour • Styles of interacting		Yes
2b	Members of the agile DWG will have a greater orientation toward appropriating technologies that support collaboration, cohesion and trust as compared to the members of the waterfall DWG.	• Instrumental uses • Advanced information technology outputs • Task outputs • Organisational environment outputs		Yes (weakly)
2c	Members of the agile DWG will state that they have more control over the quality and outcome of the product they are creating compared to members of the waterfall DWG.	• Restrictiveness • Quality		Yes (weakly)
2d	Members of the agile DWG will have more interactions between members than will the members of the waterfall DWG.	• Participation • Idea generation • Task management	2d	Yes
2e	The agile DWG will be more likely to embrace project change and make appropriations that support change compared to the waterfall DWG.	• Organisational environment • Organisational environment outputs	2e	Yes

Table 4 Summary of responses to questions for waterfall and agile DWGs

	Hypotheses	*Waterfall team*	*Agile team*
1a	The agile DWG will demonstrate structural features and spirit that differ from the waterfall DWG, given the differences in processes and values between the two methodologies.	• Restrictiveness – somewhat restrictive • Level of sophistication – very sophisticated • Comprehensiveness – very comprehensive • Decision process – team-based decisions made according to the process	• Restrictiveness – somewhat restrictive • Level of sophistication – somewhat sophisticated • Comprehensiveness – somewhat comprehensive • Decision process – consensus by team, some decisions made by upper management
1b	Members of the agile DWG will report that their structure is less hierarchical and their decision processes are less centralised than the waterfall DWG.	• Leadership – matrix and hierarchical leaning toward hierarchical • Efficiency – not efficient • Conflict management – escalation to management • Atmosphere – committed, focused, motivated	• Leadership – matrix and hierarchical leaning toward matrix • Efficiency – efficient • Conflict management – resolve within the team • Atmosphere – collaborative, fun, positive
1c	The agile DWG's internal system will be more collaborative than the waterfall DWG's internal system.	• 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
1d	Members of the agile DWG will have a more positive attitude about the agile development methodology than the members of the waterfall DWG will have about their methodology.	• 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	Members of the agile DWG will describe their atmosphere as more collaborative than will members of the waterfall DWG.	• 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 behaviour – 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 behaviour – customer and stakeholder feedback, lessons learned • Styles of interacting – collaborative partnerships, accommodating, open communication

Table 4 Summary of responses to questions for waterfall and agile DWGs (continued)

	Hypotheses	*Waterfall team*	*Agile team*
2b	Members of the agile DWG will have a greater orientation toward appropriating technologies that support collaboration, cohesion, and trust as compared to the members of the waterfall DWG.	• Instrumental uses – CMVC code repository, wikis to store project information, accept top down tool recommendation, have done video-conferencing and 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 • Organisational environment outputs – do not feel like underlying processes need to change, do not organise 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. • Organisational environment outputs - arrange teams for more communication and more co-location (teams and leaders)
2c	Members of the agile DWG will state that they have more control over the quality and outcome of the product they are creating compared to members of the waterfall DWG.	• 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
2d 2e	Members of the agile DWG will have more interactions between members than will the members of the waterfall DWG. The agile DWG will be more likely to embrace project change and make appropriations that support change compared to the waterfall DWG.	• 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 • Organisational environment – supportive, high stress, fast-paced, work and problem-solving focused • Organisational environment outputs – do not feel like underlying processes need to change, do not organise 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 • Organisational environment – collaborative, fast-paced, positive, flexible, customer-focused • Organisational environment outputs – arrange teams for more communication and more co-location (teams and leaders)

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 organised 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, Hypothesis 1a states that the agile DWG will demonstrate structural features and spirit that are unique to agile groups as compared to the waterfall DWG, given the differences in processes and values between the two methodologies. This hypothesis 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. The inter-rater agreement percentage was calculated for all nodes in all transcripts. The average agreement between nodes was 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 organised by the proposition and hypotheses that they were designed to examine.

Responses were summarised in a checklist matrix (Miles and Huberman, 1984) to understand the differences between the DWGs using the waterfall and agile methodologies (see Tables 3 and 4). The checklist matrix format was used because it helps to highlight the key differences between the waterfall and agile groups to ultimately demonstrate how group process mediates member interactions. A hypothesis was marked as 'supported' if the majority of the responses within the selected constructs supported the hypothesis and 'not supported' if the responses did not support the hypothesis. If a construct had one to two responses that did not support the hypothesis, but the overall responses still met the majority threshold, then it was noted as weakly supported.

All of our hypotheses related to Proposition 1 were fully supported and for Proposition 2, Hypotheses 2a, 2d and 2e were fully supported and Hypotheses 2b and 2c were weakly supported. We summarise the results of the hypotheses tests, as well as the evaluation of the questions in Tables 3 and 4. We discuss the findings for each set of propositions in the sub-sections that follow.

5.1 Proposition 1

The results support Proposition 1. Hypotheses 1a through 1d captured the structures within Cao et al.'s (2009) model that most influence the appropriation of the agile development methodology. These structures include:

1 agile methods defined through their structural features and spirit

2 software project characteristics

3 organisational context

4 each team's internal system including their interaction style, knowledge, expertise with agile methods and their perceptions about agile methods [Cao et al., (2009), p.334].

Differences were found between the waterfall and agile DWGs in terms of structural features and spirit, supporting Hypotheses 1a and 1b. Members of teams in the waterfall DWG were more sophisticated, comprehensive and made decisions in accordance with

the development process they were following. The members of the waterfall DWG leaned more toward a hierarchical leadership structure, as evidenced by the fact that they tended to resolve conflicts through an escalation process, although they reported that they are technically organised 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 members of the waterfall DWG did not feel that their process was efficient, they reported that their work atmosphere was one of commitment, motivation and focus. In many ways, the waterfall process had a similar structure and spirit to a military organisation.

The members of the agile DWG reported less sophistication and comprehensiveness in decision making activities and they indicated that they made their decisions through consensus. The members of the agile DWG did, however, note that some decisions were still reserved for upper management. This raises the possibility that working structures are based as much on their corporate norms as on their process choices. The members of the agile DWG reported having a matrix structure that worked in an efficient manner, with disputes and conflicts typically resolved within each group's work teams. An agile group 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."

Members felt that their atmosphere had improved from the days when the waterfall process was used and is now more collaborative, fun and positive. An agile member described his relationship with his work 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 agile group members did not differ in their feelings about the restrictiveness of their process. While almost all of the responses were consistent with the differences reported in the literature on waterfall and agile, this was one notable difference. The restrictiveness reported by members of both DWGs may be a reflection of the policies required to run a large, well-established information technology company. The company may be open to using new approaches, but they are not able to ignore legal and cross-organisational requirements, such as open-source and globalisation, that may be less applicable to smaller companies deploying agile. As with decision making, this could also be due to the history of both DWGs, which includes a shared history of using waterfall methodologies.

Hypothesis 1c argued that the internal system of an agile DWG would differ from the internal system of a waterfall DWG. The results show that this hypothesis 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. Members of the waterfall DWG reported that their work teams were more formal and it was more likely that prescriptive actions would be taken under specific circumstances. For example, one waterfall participant said …

> "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 DWGs and the nature of the products such that one backs up critical data and the other reports on the storage environment. The waterfall DWG had been using the waterfall process for many years and members apparently felt that they were subject matter experts on the waterfall process. Similarly, they reported that members of this group were competent in their ability to use the waterfall process due to their extensive experience.

The teams in the agile group had a different interaction style that valued collaborative partnerships, open communication and accommodated the needs of the group and work team. An agile group member described their meeting and 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 group's knowledge, experience and perception of others' knowledge, members of the agile DWG emphasised that they have some expertise with agile and are still in the learning phase. They understand that others may have limited experience with agile and actively try to support members who are still learning. Contrary to the waterfall group, members of the agile group consistently had positive feelings about the move to agile development. One agile 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."

Hypothesis 1d predicted that members in the agile DWG would have a more positive attitude about the agile development methodology than the waterfall group and the data support this prediction. The two constructs, persistent attitude toward appropriation and faithfulness of appropriation, were reviewed to derive the attitudes of the members of each group. For the waterfall DWG, an attitude that persistently surfaced about moving to a more agile development model was that the move would negatively impact both their product and their team. Members did not feel that they had a clear vision of how agile could be successful in their environment. They noted that during training 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. Members of the waterfall DWG said that they were able to understand where agile might benefit a work team that had different dynamics than their own, such as a small, co-located team working on a new product.

In terms of faithfulness of the appropriation, some members of the waterfall DWG had made some moves toward agile in the past, then reverted back to the traditional waterfall methodologies. When asked why the DWG reverted back, one member explained

> "I believe that given limited knowledge in agile, locality, size of the team, maybe the product, spread out, people not really realising 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.

While the agile group members also had a background in waterfall development, they had a very different attitude about agile once they had some experience and success associated with using the agile methodology. agile group members admitted that there were some challenges that they had to initially overcome with time zone differences, new tools and new ways of thinking about projects and teamwork, 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."

> "I think the interaction has increased, quite a bit, significantly. And, I think that's been a benefit."

Members of the agile DWG also felt that the additional work of moving to new tools that supported the agile process was worth the effort. One 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 members had been allowed to self-select whether they participated on the waterfall or agile project, then we might surmise that a member's personal attributes had lead them to choose the DWG using the methodology that best fit their preferences. Nevertheless, members were not able to choose which group they worked with; therefore, we speculate that the process had impacted the attitude of the members rather than the other way around.

5.2 Proposition 2

Proposition 2 argues that the appropriations by the agile DWG will create a culture that is unique from the waterfall DWG. The data show that the members of the agile DWG had a distinct culture that was different from that of the waterfall DWG, supporting Proposition 2. These findings are consistent with previous research by Doshi and Doshi (2009) that demonstrate that agile development changes the culture of the team. The results are also consistent with findings by Maznevski and Chudoba (2000) that indicate global virtual team structures also influence their culture.

Hypothesis 2a predicts that members in the agile DWG will describe their atmosphere as more collaborative than would a member of the waterfall DWG, which was supported

by the data. Members of the waterfall DWG reported that their members are committed to the project, focused on meeting project objectives and motivated to do what is needed to be successful. They reported that members are willing to support one another when assistance is needed and members are generally comfortable working with one another. When commenting on working together to achieve goals, one waterfall DWG member described his experience as follows:

> "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 members of the agile DWG shared a commitment to project success, which is a core value for their company as a whole, but they also put more focus on the relational aspects of the individual project teams and the DWG as a whole. They used the term collaborative more often and used more enthusiastic language when describing agile development and their group and teams. An agile 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 members emphasised that employees who were able to work together locally were able to create the strongest bonds between members.

Hypothesis 2b focuses on the tooling decisions made by the development groups and argues that team members in the agile group will be more likely to appropriate technologies that support collaboration, team cohesion and trust. The data supported this hypothesis, although weakly. In some cases the groups 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." Thus, one of the factors influencing tool selection and use that has not been extensively discussed in the literature is the history of the agile team and constraints presented by the organisation's standards for software adoption. Waterfall and agile groups that work for different companies and are given complete control to decide on the tools that are best for their projects, would likely make different technology choices.

In general, members in both DWGs used many of the same tools within their groups. This is likely because they both work for the same company and because the company dictates many of the tools that they must use. Members of both groups regularly used phone, e-mail, instant messaging, screen sharing and wikis. Team members who were part of the waterfall DWG tried to use video conferencing, but came to the conclusion that it was not beneficial. The primary difference between the two groups is that members of the waterfall DWG were resistant to moving to new tools, while members of the agile group embraced and even pursued changes that would include using new tools. The agile group members made a concerted effort to use more agile-friendly tools that they learned were appropriate for agile development; 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 members maintained a social wiki that contained a weekly newsletter with articles about team members to encourage bonding. The challenge the agile members faced were not related to training and initiative, but rather from organisational history and constraints imposed by the organisation's standards for software adoption.

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

Team members in the agile DWG also felt confident that they were creating a product based on the needs of the customer, whereas the waterfall group members could not be as reactive to customer requests and focused more rigidly on delivering the product that the company roadmap requested. One waterfall 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."

Hypothesis 2d predicts that members of the agile DWG will have more interactions than members in the waterfall group and the data support this hypothesis. Members, teams and subgroups in the waterfall DWG held weekly meetings to review project status in a formal manner using charts and assigned presenters. Members interacted one-on-one or in smaller groups, but those interactions tended to be ad hoc, only when interactions were needed. When asked about interactions, one waterfall 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, agile members focused heavily on team interactions. Their use of sprints to manage project cycles keeps the members working at a fast pace and daily Scrum meetings are a key part of their success. They reorganised the global group members so the individual 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. Group members still regularly communicate with other

members in different time zones, but the core team members are available during similar work hours. One agile 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 members reported that the increased interaction with other members positively impacted the overall project.

The final Hypothesis, 2e, states that members of agile DWGs will be more likely to embrace project change and make appropriations that support change compared to members of the waterfall DWG. This hypothesis was also supported by the data. Members of teams in the waterfall DWG were not completely adverse to change, but they were more resistant to making changes compared to agile members. They felt that they had a solid, repeatable process that worked for them and process changes introduced unnecessary risk to the project. One waterfall 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."

Team members in the waterfall DWG also avoided changing tools and indicated that changing to a new tool would be a major investment in terms of training, moving project code and data and creating new templates. They admitted that they were aware of available tools that could help their project, but they did not feel 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 then they reported they would often seek an exception.

Members of the agile DWG had a different attitude toward change. They reported that change had generated extra work for their members in terms of training, tool migration and adjusting to new interaction styles, but they felt that short-term setbacks were worth the effort given the overall benefit of the product produced because of the changes in methodology brought by agile. The agile DWG had been one of the first units in their area in the company to try agile development and they promoted the benefits of agile to their management team so they could continue to use it in a more robust way with each release. Members proactively took the initiative to try out new tools that were more supportive of the agile methodology. One agile 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 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 group 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 on a regular basis. This coach was also available to the waterfall DWG, but members generally chose not to work with him. After each sprint, agile teams held retrospectives or lessons learned sessions and adjusted their process and operations based on the feedback.

6 Discussion

Research by the Project Management Institute (PMI) found that the use of agile methodologies tripled from December of 2008 to May of 2011 and many subsequent industry surveys suggest that the move to agile is only accelerating (Geracie et al., 2012). Such predictions emphasise the need for better understanding about the impact of agile methodologies impact the work environment and so in this study, we set out to examine how development methodologies influence team work structures and culture.

Our research focused on two propositions. The first proposition was that the DWG using the agile methodology would demonstrate real differences in structures, spirit and attitudes when compared to waterfall DWGs. The hypotheses associated with this first proposition were all supported, demonstrating that there were clear differences between the DWGs and that the agile DWG reported less hierarchy, more use of collaborative systems and had a greater confidence in the agile method. The second proposition posited that the DWG that adopted agile would have a culture unique and distinct from the waterfall DWG. The hypotheses supporting this proposition were also supported, meaning that the agile DWG members felt more collaborative, more interactive and appropriated technologies to empower these values. The agile DWG also reported that they had more control over quality and felt more accepting of project change.

The results from our research show that agile and waterfall methodologies engender different work structures and culture, beyond what the methodologies themselves require. Additionally, whereas Cao et al. (2009) only studied agile development teams, our research substantially extends their work by examining both waterfall and agile teams. This is an important contribution, as we are aware of no prior research that has compared agile and waterfall software development groups in the same organisation. Thus, our research not only provides support for work done by Cao et al. (2009), but by comparing the agile team directly with a waterfall team in the same organisation, we demonstrate conclusively that effects seen in an agile context are a function of the agile methodology and not an artefact of extant organisational structures and culture. Thus, by examining established DWG's, both of which had been encouraged to explore agile, we respond to the call by Abrahamsson et al. (2009) to understand agile effects beyond the adoption stage.

This research also builds on previous team research (e.g., Martins et al., 2004) by demonstrating that the process, or more generally, the organisational rules that a group or team is following will impact their interactions. In particular, group and team members will not necessarily interact because their technology allows them to do so or because they have been assigned to the same team. Rather, the framework they are working within will guide the amount of interaction and the nature of that interaction, providing of ecology for the team's subsequent evolution. In our study, we examined two groups that both have the same communicative infrastructure. In spite of these affordances' availability, the members in the waterfall DWG (compared to the agile DWG) interacted less and their communications were more formal simply because this is the nature of working under the processes laid out in the waterfall methodology. The groups that decided to use agile interacted daily, created core work teams in similar time zones and sought tools that would facilitate rich and frequent collaboration with team members. Our research clearly demonstrates that methodology choice influenced the teams' work structures and culture beyond the requirements of the methodology; in other words, agile DWG members' greater feelings of collaboration and confidence in their development

methodology are a derivative effect of the implementation of agile. While the agile methodology does require some forms of affective change; our research suggests that its implementation also provided an ecology within which these non-required changes would evolve.

In our study, all of the participants were from the same organisation, which offers a degree of control that is important in parsing out the role of the development methodology each group used. Prior research examining development methodologies has generally involved either a case study of one organisation or comparisons of agile and waterfall teams between organisations (Mangalaraj et al., 2009). Because our informants all came from the same organisation, their shared history with the organisation, its values and its culture suggest that the differences that we do see reflect more on the nature of agile and waterfall processes and philosophies than on factors that might be idiosyncratically associated with different organisations. For example, the similarities in restrictiveness and control over quality and outcomes in this study is likely the result of the employees sharing a common perspective derivative from the culture and perspectives espoused by the leadership of their organisation. Nevertheless, further research is needed involving teams within other organisations using both waterfall and agile methodologies to validate these findings. Research shows that waterfall and agile methodologies are sometimes deployed differently between small and large organisations (Kahkonen, 2004; Lindvall et al., 2004), so including teams from firms of various sizes would be a logical extension of this research stream. Additionally, future research could consider the ways that different teams interpret agile methods and how these differing interpretations influence their team structure, processes and tool sets.

The agile DWG in the present study also had a 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 organisations that have less-experienced teams as well as experienced teams to better understand if experience level impacts attitudes toward change. Similarly, studies have shown differences in global and non-global teams (Jarvenpaa and Leidner, 1999; Maznevski and Chudoba, 2000), emphasising the need for both types of teams to be included in future studies.

The findings from the study highlight differences in team dynamics between waterfall and agile teams and emphasise the impact of process choice on teams. A software development team moving to agile will need to consider cultural, tooling and attitude shifts that need to occur to make a successful transition. These challenges become more pronounced when a DWG is composed of both virtual and co-located members (Korkala et al., 2009; Sarker and Sarker, 2009). For example, a DWG with members distributed globally may have to change its structure to support daily interactions by, for example, coordinating schedules and managing time-zone differences. When a team is using the agile methodology, members will need to adapt to new tools supporting the fast-paced, highly interactive nature of agile development as well as change their communication behaviours to accommodate the more frequent interactions with team members and clients. Given the challenges a global DWG has when appropriating agile, our results suggest that proximity is important; therefore, managers should seek to co-locate team members to the greatest degree possible. Our results align with previous research that suggests that collocated teams are the most ideal structure for agile development because

informants indicated that collocated teams embraced the agile methodology more quickly than did virtual teams (Law and Ho, 2004).

Additionally, our results suggest that care must be taken in moving teams to agile or agile-like methodologies. In other words, not all task-process-team structure combinations are ideal, particularly when teams are asked to engage with new methodologies or tools. For example, we found that, given the choice, waterfall DWGs retained waterfall methodologies even though organisational pressure and demonstrated success with agile in other areas of the organisation suggested that these newer alternatives might result in improved processes or outcomes. This team's decision was based, in part, on evaluations of reward versus risk made by team members and project leads that were utilising methodologies (i.e., waterfall) that had worked in the past (e.g., our current processes work and the task of shifting resources would be difficult). This finding is important in guiding team leaders and managers considering a change to agile teams.

7 Limitations

As with any change in software or process, there is a switching cost that a team incurs when transitioning to a new methodology; thus, as has been routinely done in research examining technology adoption, future research should examine the nature of the real and perceived costs of changing methodologies and how these costs influence the decision making process associated with the appropriation of agile (Sarker and Sarker, 2009).

As noted above, much of the prior research has focused on successful transitions to agile as well as guidelines for making the transition (e.g., Chow and Cao, 2008; Conboy et al., 2011; Procaccino et al., 2002; Fitzgerald et al., 2006; Hosalkar and Bowonder, 2000; Meso and Jain, 2006; Misra et al., 2009; Nielsen and McMunn, 2005; Nerur et al., 2005; Sarker and Sarker, 2009; Srinivasan and Lundqvist, 2009; Tan and Teo, 2007). Our research did not focus on the adoption process, but rather on the current operation of two large groups using different project methodologies. However, in the process of creating a retrospective on the groups' histories, our research uncovered the factors (project complexity combined with a perceived lack of benefit from customer participation) that influenced one group to stay with traditional methodologies, despite significant organisational pressure to adopt agile. This underscores the contextual ecology that adoption depends upon and that for this group, led them to retain the waterfall methodology.

There are many methodologies that fall under the agile umbrella (Ashmore and Runyan, 2014). The company we interviewed primarily used scrum and EP. Future studies should explore other agile methods, such as Kanban or Crystal and should be investigated across a wider variety of company types (e.g., large corporations and start ups).

8 Conclusions

In summary, this study makes several important contributions to research on teams, software development and AST. First, we found that the process a team deploys does mediate the dynamics of a team. Specifically, we found that, when compared to agile,

waterfall DWGs interacted less and the communication acts were more formal. This was because these DWGs had used the waterfall process and followed the prescripts for managing projects that align with this technique. Members of agile DWGs generally interacted daily, created core teams in similar time zones and sought tools that would facilitate collaboration with among members because the agile methodology specifically calls for frequent, interactive dialog about project activities. While this is not surprising, it does demonstrate that the methodology adopted by a team has important influences not only on logistical team processes, but also on communication style, activities and, in the end, attitudes. Additionally, this study is the first that we are aware of that has directly compared waterfall and agile development groups in the same organisation. This is important because the majority of software development research has examined either waterfall or agile development teams, but not both types of teams in the same organisational context.

This study compares the differences between the two teams in terms of structural, cultural and technological differences. Findings suggest that there are important differences between the two types of groups that need to be considered by members and their leadership teams. Specifically, when teams adopt agile they need to be prepared to change many facets of their work activities, which implies that team members need to be supported in these changes with training, technological support and time for adaptation. While many of these findings would appear to be obvious in light of comparing the features of the two methodologies, this study confirms that the differences between work activities and outcomes are driven largely by the methodologies and not by other organisational or cultural differences that have been present in prior research studying these methodologies.

References

Abrahamsson, P., Conboy, K. and Wang, X. (2009) ''Lots done, more to do': the current state of agile systems development research', *European Journal of Information Systems*, Vol. 18, No. 4, pp.281–284.

Ashmore, S. and Runyan, K. (2014) *Introduction to Agile Methods*, Addison-Wesley Professional, Upper Saddle River, NJ.

Ashmore, S. and Wedlake, M. (2016) 'Developing the product your customer really wants: the value of an agile partnership', *Information Resources Management Journal (IRMJ)*, Vol. 29, No. 3, pp.1–11.

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*, Addison-Wesley, Boston.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. and Thomas, D. (2001) *Manifesto for Agile Software Development* [online] http://www.agilemanifesto.org/, (accessed 17 November 2014).

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* [online] http://www.ibm.com/developerworks/rational/library/4029.html, (accessed 1 May 2011).

Cao, L., Mohan, K., Xu, P. and Ramesh, B. (2009) 'A framework for adapting agile development methodologies', *European Journal of Information Systems*, Vol. 18, No. 4, pp.332–343.

Chow, T. and Cao, D-B. (2008) 'A survey study of critical success factors in agile software projects', *Journal of Systems and Software*, Vol. 81, No. 6, pp.961–971.

Cockburn, A. and Highsmith, J. (2001) 'Agile software development: the business of innovation', *Computer*, Vol. 34, No. 9, pp.120–127.

Conboy, K. (2009) 'Agility from first principles: reconstructing the concept of agility in information systems development', *Information Systems Research*, Vol. 20, No. 3, pp.329–354.

Conboy, K., Coyle, S., Wang, X. Pikkarainen, M. (2011) 'People over process: key challenges in agile development, *Software*, Vol. 28,No. 4,pp.48–57, IEEE.

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

DeSanctis, G. and Poole, M.S. (1994) 'Capturing the complexity in advanced technology use: adaptive structuration theory', *Organization Science*, Vol. 5, No. 2, pp.121–147.

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

Düchting, M., Zimmermann, D. and Nebe, K. (2007) 'Incorporating user centered requirement engineering into agile software development', in Jacko, J.A. (Ed.): *Human-Computer Interaction, Interaction Design and Usability, HCI 2007, Lecture Notes in Computer Science*, Vol. 4550, Springer, Berlin, Heidelberg.

Fiol, C.M. and O'Connor, E.J. (2005) 'Identification in face-to-face, hybrid and pure virtual teams: untangling the contradictions', *Organization Science*, Vol. 16, No. 1, pp.19–32.

Fitzgerald, B., Hartnett, G. and Conboy, K. (2006) 'Customising agile methods to software practices', *European Journal of Information Systems*, Vol. 15, No. 2, pp.200–213.

Fowler, F.J. (1995) *Improving Survey Questions*, Vol. 38, Sage, Thousand Oaks.

Geracie, G., Heidt, D. and Starke, S. (2012) *The Study of Product Team Performance*, Actuation Consulting, Inc., [online] http://www.actuationconsultingllc.com/buy/Study of_Product_Team_Performance_2012.pdf (accessed 14 May 2014).

Giddens, A. (1977) *Studies in Social and Political Theory*, Basic Books, New York.

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

Grimaldi, P., Perrotta, L., Corvello, V. and Verteramo, S. (2016) 'An agile, measurable and scalable approach to deliver software applications in a large enterprise', *International Journal of Agile Systems and Management*, Vol. 9, No. 4, pp.326–339.

Guzzo, R.A. and Dickson, M.W. (1996) 'Teams in organizations: recent research on performance and effectiveness', *Annual Review of Psychology*, Vol. 47, No. 1, pp.307–338.

Hosalkar, A. and Bowonder, B. (2000) 'Software development management: critical success factors', *International Journal of Technology Management*, Vol. 19, No. 7, pp.760–772.

Jarvenpaa, S. and Leidner, D. (1999) 'Communication and trust in global virtual teams', *Organization Science*, Vol. 10, No. 6, pp.791–815.

Kahkonen, T. (2004) 'Agile methods for large organizations – building communities of practice', Paper presented at the *Agile Development Conference*.

Katzenbach, J.R. and Smith, D.K. (1993) 'The discipline of teams', *Harvard Business Review*, March–April, Vol. 71, pp.111–146.

Korkala, M., Pikkarainen, M. and Conboy, K. (2009) 'Distributed agile development: a case study of customer communication challenges', in *Proceedings of the 10th International Conference, XP 2009*, Pula, Sardinia, Italy, 25–29 May, pp.161–167.

Larman, C. (2004) *Agile and Iterative Development: A Manager's Guide*, Pearson Education, Boston, MA.

Larman, C. and Basili, V.R. (2003) 'Iterative and incremental development: a brief history', *Computer*, Vol. 36, No. 6, pp.47–56, DOI: 10.1109/mc.2003.1204375.

Law, A. and Ho, A. (2004) 'A study case: evolution of co-location and planning strategy', Paper presented at the *Proceedings of the Agile Development Conference*.

Lee, E.C. (2008) Forming to performing: transitioning large-scale project into agile, *AGILE Conference 2008*, pp.106–111.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D. and Kahkonen, T. (2004) 'Agile software development in large organizations', *Computer*, Vol. 37, No. 12, pp.26 –34.

Majchrzak, A., Rice, R.E., Malhotra, A., King, N. and Ba, S. (2000) 'Technology adaptation: the case of a computer-supported inter-organizational virtual team', *MIS Quarterly*, Vol. 24, No. 4, pp.569–600.

Mangalaraj, G., Mahapatra, R. and Nerur, S. (2009) 'Acceptance of software process innovations – the case of extreme programming', *European Journal of Information Systems*, Vol. 18, No. 4, pp.344–354.

Martins, L., Gilson, L. and Maynard, M.T. (2004) 'Virtual teams: what do we know and where do we go from here?', *Journal of Management*, Vol. 30, No. 6, pp.805–835.

Maznevski, M.L. and Chudoba, K.M. (2000) 'Bridging space over time: global virtual team dynamics and effectiveness', *Organization Science*, Vol. 11, No. 5, pp.473–492, DOI: 10.1287/orsc.11.5.473.15200

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

Meso, P. and Jain, R. (2006) 'Agile software development: adaptive systems principles and best practices', *Information Systems Management*, Vol. 23, No. 3, pp.19–30.

Miles, M. and Huberman, A.M. (1984) *Qualitative Data Analysis: A Sourcebook of New Methods*, Sage, Beverly Hills.

Misra, S.C., Kumar, V. and Kumar, U. (2009) 'Identifying some important success factors in adopting agile software development practices', *Journal of Systems and Software*, Vol. 82, No. 11, pp.1869–1890.

Misra, S.C., Kumar, V., Kumar, U. and Grant, G. (2006) 'The organizational changes required and the challenges involved in adopting agile methodologies in traditional software development organizations', *1st International Conference on Digital Information Management*, December, pp.25–28.

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

Murphy, T.E., Duggan, J., Norton, D., Prentice, B., Plummer, D.C. and Landry, S. (2009) *Predicts 2010: Agile and Cloud Impact Application Development Directions*, edited by G. Research, Stamford, CT.

Nerur, S., Mahapatra, R. and Mangalaraj, G. (2005) 'Challenges of migrating to agile methodologies', *Communications of the ACM*, Vol. 48,No. 5, pp.72–78.

Nielsen J. and McMunn D (2005) 'The agile journey: adopting XP in a large financial services organisation', in *Proceedings of the 6th International Conference on eXtreme Programming and Agile Processes*, Sheffield, UK, 18–23 June, pp.28–37.

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

Procaccino, J.D., Verner, J.M., Overmyer, S.P. and Darter, M.E. (2002) 'Case study: factors for early prediction of software development success', *Information and Software Technology*, Vol. 44, No. 1, pp.53–62.

Project Management Institute (2014) *The World is Quickly Becoming Agile. Are You?* [online] http://www.pmi.org/ (accessed 20 February 2014).

Raccoon, L.B.S. (1997) 'Fifty years of progress in software engineering', *SIGSOFT Software Engineering Notes*, Vol. 22, No. 1, pp.88–104.

Raghuram, S., Tuertscher, P. and Garud, R. (2010) 'Research note---mapping the field of virtual work: a cocitation analysis', *Info. Sys. Research*, Vol. 21, No. 4, pp.983–999.

Ramesh, B., Cao, L., Mohan, K. and Xu, P. (2006) 'Can distributed software development be agile?', *Commun. ACM*, Vol. 49, No. 10, pp.41–46.

Rohunen, A., Rodriguez, P., Kuvaja, P., Krzanik, L. and Markkula, J. (2010) 'Approaches to agile adoption in large settings: a comparison of the results from a literature analysis and an industrial inventory', in *Proceedings of the 11th International Conference PROFES 2010*, Limerick, Ireland, 21–23 June, pp.77–91.

Sarker, S. and Sarker, S. (2009) 'Exploring agility in distributed information systems development teams: an interpretive study in an offshoring context', *Information Systems Research*, Vol. 20, No. 3, pp.440–461.

Sawyer, S. (2004) 'Software development teams', *Communication of the ACM*, Vol. 47, No. 12, pp.95–99, DOI: 10.1145/1035134.1035140.

Schwaber, K. and Beedle, M. (2002) *Agile Software Development with Scrum*, Prentice Hall, Upper Saddle River, NJ.

Scott, M. and DeSanctis, P.G. (1992) 'Microlevel structuration in computer-supported group decision making', *Human Communication Research*, Vol. 19, No. 1, pp.5–49, DOI: 10.1111/j.1468-2958.1992.tb00294.

Sidky, A., Arthur, J. and Bohner, S., (2007) 'A disciplined approach to adopting agile practices: the agile adoption framework', *Innovations in Systems and Software Engineering*, Vol. 3, No. 3, pp.203–216.

Singh, A., Singh, K. and Sharma, N. (2015) 'Agile in global software engineering: an exploratory experience', *Int. J. of Agile Systems and Management*, Vol. 8, No. 1, pp.23–38.

Srinivasan, J. and Lundqvist, K. (2009) 'Organizational enablers for agile adoption: learning from GameDevCo', in *Proceedings of the 10th International Conference, XP 2009*, Pula, Sardinia, Italy, 25–29 May, pp.63–72.

Tan, C.H. and Teo, H.H. (2007) 'Training future software developers to acquire agile development skills', *Communications of the ACM*, Vol. 50, No. 12, pp.97–98.

Warkentin, M. and Beranek, P.M. (1999) 'Training to improve virtual team communication', *Information Systems Journal*, Vol. 9, No. 4, pp.271–289, DOI: 10.1046/j.1365-2575. 1999.00065.x.

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

Williams, L. and Cockburn, A. (2003) 'Guest editors' introduction: agile software development: it's about feedback and change', *Computer*, Vol. 36, No. 6, pp.39–43, DOI: 10.1109/mc.2003.1204373.

Wilson, M. (2013) *Welcome to the Post Waterfall Era*, Gartner Blog Post, [online] http://blogs.gartner.com/nathan-wilson/welcome-to-the-post-waterfall-era/ (accessed 24 April 2014).

Woodward, E., Surdeck, S. and Ganis, M. (2010) *A Practical Guide to Distributed Scrum*, Pearson Education, Inc., Boston, MA.

Yin, R.K. (2009) *Case Study Research: Design and Methods*, Vol. 5, Sage, Los Angeles, CA.

Notes

1 We use the term 'development work group' because of the size of these units (i.e., approaching or exceeding 100 members). Guzzo and Dickson (1996) suggest that a *work group* "is made up of individuals who see themselves and who are seen by others as a social entity, who are interdependent because of the tasks they perform as members of a group, who are embedded in one or more larger social systems (e.g., community, organisation), and who perform tasks that affect others (such as customers or co-workers)." (p.308). We added the adjective 'development' because it encompasses the shared goals of IT development work. When we reference teams, we are discussing smaller units within each DWG [see Katzenbach and Smith (1993) for a discussion of group versus team size]. These two *groups* worked on two large projects that each included numerous tasks and subtasks and, as a result, in the agile DWG, numerous teams worked on each project. A project team consists of anywhere from six to 19 members who may be distributed globally.

Appendix A

Interview questions

Group #: _____

Demographic/cntext:

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 US?

10 How many team members do you communicate with regularly who reside outside the US?

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

Sources of advanced information technology:

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

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

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

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

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

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

18 How do you resolve conflicts on your project?

19 How would you describe the atmosphere of your project?

Other sources of structure:

20 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)?

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

Emergent sources of structure:

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

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

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

New social structures:

25 How are rules created and modified?

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

Social interaction:

27 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?

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

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

30 How are these tools selected by your team?

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

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

33 How are ideas generated on the team?

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

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

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

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

Group's internal system:

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

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

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

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

Decision outcomes:

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

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

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

45 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 5 for reference during the coding.

Table 5 AST nodes for coding

Node	*Node colour*	*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?
Idea generation	Red	How are ideas generated by the team?
Influence behaviour	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?
Organisation environment	Green	What is the organisational environment? Examples could be positive, negative, collaborative, fast-paced, or something similar that describes how the team works.
Organisation 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 the tasks are 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 tasks are 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?