

# **How effective are Agile Methods for Distributed Software Development?**

**Pat Mulhern**

A dissertation submitted to the University of Dublin  
in partial fulfilment of the requirements for the degree of  
MSc in Management of Information Systems

***1<sup>st</sup> September 2015***

## Declaration

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work, and has not been submitted as an exercise for a degree at this or any other university. I further declare that this research has been carried out in full compliance with the ethical research requirements of the School of Computer Science and Statistics.

Signed: \_\_\_\_\_

Pat Mulhern

1<sup>st</sup> September 2015

## **Permission to lend and/or copy**

I agree that the School of Computer Science and Statistics,  
Trinity College may lend or copy this dissertation upon  
request.

Signed: \_\_\_\_\_

Pat Mulhern

1<sup>st</sup> September 2015

## **Acknowledgements**

I would like to thank Ms. Maria Maloney for her time and expertise in helping me complete this dissertation.

Thanks also to the lecturers, staff and my classmates in Trinity College Dublin.

Finally to my wife Karina and sons Jack, Tom and Max for their ongoing support and understanding throughout the past two years.

## **Abstract**

Today organisations trade in an environment where change is constantly forcing software developers to be nimble, responsive and incorporate design changes on a daily basis. These methods encourage development teams to embrace change by focusing on adaptability over planning. Distributed Software Development (DSD) is also becoming more common in order to reduce labour costs and reduce the time to market. However, research shows that time zones and geographical distance in DSD can affect communication and collaboration, which adversely affects the project. This research discusses the challenges faced by organisations using agile methods with DSD teams. It takes an interpretive approach to explore the interactions between people, process, geographical/temporal distance and cultural influences. The method of data collection used for this research is semi-structured interviews and is qualitative in nature. The Constant Comparative Method is used to analyse the data. The study reveals that the values of agile software development such as an emphasis on face-to-face communication, contact with customers and informal communications are challenged when DSD teams are used. Finally, the research presents a framework, entitled the 'Hybrid DSD Lifecycle', to address the challenges of DSD, which uses a combination of agile, and waterfall methods to best suit the DSD environment.

---

## Table of Contents

1	INTRODUCTION.....	1
1.1	Background.....	1
1.2	Research Question .....	1
1.3	Research Scope.....	2
1.4	Beneficiaries of the Research.....	2
1.5	Dissertation Layout .....	3
2	LITERATURE REVIEW .....	4
2.1	Introduction .....	4
2.2	History of Software Development .....	4
2.3	What are Agile Methods? .....	8
2.4	What is DSD? .....	8
2.5	Benefits and challenges of agile methods and DSD .....	9
2.6	Summary of Literature Review .....	15
2.7	Conclusion .....	23
3	METHODOLOGY AND FIELDWORK .....	24
3.1	Introduction .....	24
3.2	Research Philosophy .....	24
3.3	Research Approach .....	27
3.4	Research Strategies Considered.....	29
3.5	Research Choices.....	30
3.6	The Question Formulation Process .....	31
3.7	Data Collection.....	31
3.8	Research Ethics .....	32
3.9	Lessons Learned.....	33
3.10	Conclusion .....	33
4	FINDINGS AND ANALYSIS .....	34
4.1	Introduction .....	34
4.2	Participant Profile .....	34
4.3	Industrial Sector .....	35

---

4.4	Data Analysis .....	36
4.5	Interview Findings .....	41
4.6	Summary of Findings .....	52
5	CONCLUSIONS AND FUTURE WORK .....	59
5.1	Introduction .....	59
5.2	Answering the research questions .....	59
5.3	Limitations of the Research .....	62
5.4	Recommendations for Future Work.....	62
5.5	Conclusion .....	63
<b>6</b>	<b>References.....</b>	<b>64</b>
	Appendices .....	68
	Appendix 1 – Ethics Application.....	68
	Appendix 2 – Information Page for Participants .....	70
	Appendix 3 – Informed Consent Form .....	72
	Appendix 4 - Interview Questions .....	75

## List of Figures

- FIGURE 2.1 - Build and Fix Approach (Dawson 2009)
- FIGURE 2.2 - Waterfall Model (Royce 1970)
- FIGURE 2.3 - Iterative Model (Dawson 2009)
- FIGURE 3.1 - Research Onion (Saunders, 2009)
- FIGURE 3.2 - Deductive Research Approach (Trochim 2006)
- FIGURE 3.3 - Inductive Research Approach (Trochim 2006)
- FIGURE 4.1 - Participants Industrial Sector
- FIGURE 4.2 - Imported Transcripts
- FIGURE 4.3 - Codification of Keywords within the Transcripts
- FIGURE 4.4 - Code Cloud taken from the Semi-Structured Interviews
- FIGURE 4.5 - Initial Codification of Transcripts.
- FIGURE 4.6 - Code Co-Occurrence
- FIGURE 4.7 - Effectiveness of Agile Methods
- FIGURE 4.8 - Hybrid DSD Lifecycle
- FIGURE 5.1 - Hybrid DSD Lifecycle

## List of Tables

- TABLE 2.1 - Software Development Life Cycle (Royce 1970)
- TABLE 2.2 - Literature reviewed for this study and findings
- TABLE 3.1 - Comparison of Four Research Philosophies Saunders et al. (2009)
- TABLE 4.1 - Participant Profiles
- TABLE 4.2 - Code Frequency
- TABLE 4.3 - Summary of Research Findings
- TABLE 4.4 - DSD Implementation Framework



## List of Abbreviations

CAQDAS	Computer Assisted Qualitative Data Analysis Software
CMM	Capability Maturity Model
DSD	Distributed Software Development
SDLC	Software Development Lifecycle
U.S.	United States
U.K.	United Kingdom
V.P.	Vice President
XP	Extreme Programming

## **1 INTRODUCTION**

The focus of this research study centers on distributed software development (DSD) and agile methods used for software development and project management. Persson et al. (2012) states that agile and DSD combined has emerged as a significant trend for organisations engaged in software development over the past ten years. This research investigates the effectiveness of agile methods for DSD.

### **1.1 Background**

In 2001, the Agile Manifesto was developed in recognition of the need to change the way software was developed (Highsmith et al. 2001). According to Highsmith et al. (2001), the changing business environment also affects the software development processes. They state that the customer must be satisfied at the time of delivery instead of the project initiation. Highsmith et al. (2001) states that software development needs procedures that not so much deal with how to stop change early in a project, but how to better handle changes throughout its life cycle. The Agile Manifesto was developed to embrace these changes during development. Agile is suitable for colocation where developments teams sit in the same physical location and coordinate tasks through face-to-face interaction, with minimal requirements documentation and informal knowledge transfer (Highsmith 2002).

Ramesh et al. (2012) describes distributed development as information systems development in which project stakeholders are dispersed along the following boundaries: geographical, organisational and temporal. Research conducted by Prikladnicki & Audy (2012) states that organisations are using DSD to take advantage of lower cost labour, access to skilled human resources, flexibility, and competitive advantages. Conchúir et al. (2001) state that organisations reduce costs by moving software development to low wage countries. DSD also introduces a number of challenges in relation to communication, coordination and control of the development process caused by geographical, time difference and cultural issues (Conchúir et al. 2001).

### **1.2 Research Question**

The goal of this research is to understand, review and propose solutions to these challenges associated with agile methods and DSD. Additionally, the research will explore how organisations are currently implementing agile methods for DSD and understand their successes and failures.

- The primary research question posed in this study is:

How effective are agile methods for distributed software development?

The question gives rise to the following sub questions:

- What factors influence the choice of using waterfall based methods over agile methods in distributed software development?
- What factors make specific projects more suited to agile development in distributed software development than others?
- Is there a framework that can help organisations better manage distributed software development?

### **1.3 Research Scope**

The research targets ten organisations that are currently using agile methods for DSD. These organisations were selected because of their experience and ability to provide information that is pertinent and appropriate to this research. All participants are working for organisations with DSD teams in two or more geographic locations. Ten semi-structured interviews were conducted for this research, with all participants being senior managers and decision makers within the DSD function of their respective organisations. Typically, the experience of those interviewed ranged from - project managers, team leaders, software architects, line managers, senior software developers, system developers and scrummasters. The participants are located in Ireland, India, U.K. and the U.S.

### **1.4 Beneficiaries of the Research**

Organisations are using agile methodologies to manage, develop and deliver software projects with DSD teams. According to Vijayaraghavan et al. (2014) the Capability Maturity Model (CMM) operates successfully in local environments but it does not adequately address the challenges of DSD. Cohen et al. (2004) writing about agile methods explains CMM as a five-level model that describes good engineering and management practices and prescribes improvement priorities for software organisations. At present, no dedicated global standard exists for agile in DSD. However, agile software development methods have been adapted to manage DSD projects. For example, Vijayaraghavan et al. (2014) states that in order to address the challenges arising from DSD, large team size and lack of co-location, practices from traditional sequential development methods are integrated into agile methods to form a hybrid model.

Research in this area will help to further the knowledge and expertise that already exists in relation to agile and DSD. This research seeks to investigate what is currently

happening in the agile and DSD communities today and to provide a roadmap for further research in this area.

## **1.5 Dissertation Layout**

The dissertation consists of five chapters. An overview of each chapter is given below.

Chapter 1: Introduction: This chapter presents the background information in relation to the research question and scope. Its purpose is to introduce the reader to the topic and the challenge the dissertation investigates. It sets the scene for the research project.

Chapter 2: This chapter goes into more detail about the subject matter of the dissertation. The Literature Review critically analyses available literature on the effectiveness of agile methods for (DSD). It also, provides a deeper understanding of how organisations currently adapt agile methods with DSD to suit their needs.

Chapter 3: The research Methodology and Field Work presents the research approaches considered, the chosen methodology and the rationale for the chosen method. It then looks at the limitations of the research and concludes with the ethics implementation and the lessons learned from conducting the research project.

Chapter 4: Findings and Analysis explains how the data is collected and analysed. It provides a summary of the findings.

Chapter 5: Conclusions and Future work discusses how the findings of the study answers the research question and sub-questions. Also discussed are the key findings, new and interesting findings. The chapter concludes with the limitations of the research and recommendations for future research in this area.

## 2 LITERATURE REVIEW

### 2.1 Introduction

The purpose of this chapter is to review the literature on the effectiveness of agile methods for distributed software development (DSD). It also provides a deeper understanding of how organisations adapt agile methods with DSD to suit their needs. Section 2.2 explores the history of software over the last 50 years and the evolution of software development life cycles. The agile development method is presented in Section 2.3. Section 2.4 describes DSD. Section 2.5 discusses the current research and highlights benefits and challenges of agile methods and DSD combined. Section 2.6 summarises and concludes the chapter.

### 2.2 History of Software Development

According to Dawson (2009) software was first developed in the mid twentieth century in an ad hoc way without a defined development process. Royce (1970) identified the need to address the challenge of organising software development activities into a meaningful method or Software Development Life Cycle. A Software Development Life Cycle (SDLC) is a collection of phases used to develop an information system (Fitzgerald & Avison 2003). Boehm (1988) argues that all software development projects will progress through a series of stages. The five stages and the purpose of each stage according to Royce (1970) are shown in Table 2.1.

TABLE 2.1 - Software Development Life Cycle (Royce 1970)

Stage	Description
Requirements	The requirements stage documents the problem to be solved and associated requirements in the environment in which the system will function.
Design	The design stage represents the software design based on the requirements. The requirement is broken down into modules and the interfaces between the modules are defined.
Build	The build stage is the coding of the software system where code development and walkthrough is done to ensure the code meets the requirements.
Test	The test stage is the testing of the code to ensure the code fits together and performs as designed.

Stage	Description
Implementation	The implementation stage is the installation and acceptance of the software by users based on the requirements collected earlier. This stage involves fixing of any bugs found by the end users.

Before discussing agile methods for DSD, it is important to understand the history of software development methods over the past 50 Mohammed et al. (2010) and Dawson (2009) identify the most common software development methods as follows:

- The Build and Fix Method
- Stage-wise/Waterfall model
- Incremental/ Iterative Method
- Agile

According to Dawson (2009) the build and fix method as shown in Figure 2.1 was used in the pioneering days of software development. Boehm (1988) states the build and fix model consisted of two steps:

- Write code
- Fix the code problem

Figure 2.1 demonstrates the process where programmers would write some code, run the code and then correct any bugs in the software. Boehm (1988) reveals that this approach lead to poorly structured code, high maintenance costs and did not meet the end user's needs.

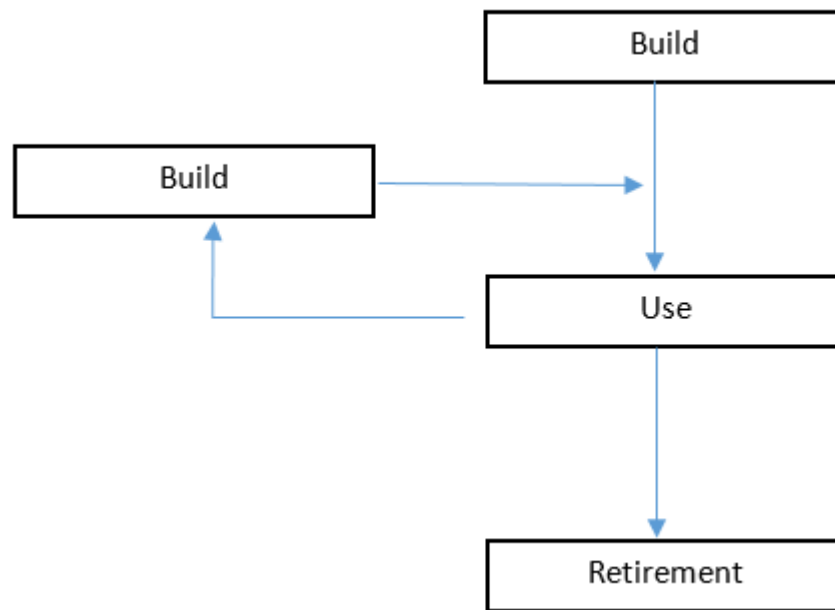


FIGURE 2.1 - Build and Fix Approach (Dawson 2009)

According to Boehm (1988) the stage-wise approach succeeded the build and fix approach. Boehm (1988) reveals that the stage-wise model mandated that software development should follow a series of successive stages. Following the stage-wise model, Royce (1970) defined the waterfall model, thus refining further the stage-wise approach. The waterfall model is a series of sequential activities leading from requirements capture through to the implementation of the system. The study identified requirements, design, coding, testing and testing as sequential SDLC stages in the waterfall model. According to Mohammed et al. (2010), waterfall consists of several non-overlapping stages as shown in Figure 2.2. The output of each phase is the input to the next phase. Mohammed et al. (2010) argues that one of the big disadvantages of waterfall is that customers are frustrated by the long lead-time from specification of requirements through to getting access to working software. Boehm (1988) reveals that waterfall is unsuited to projects where interactive end users applications are developed.

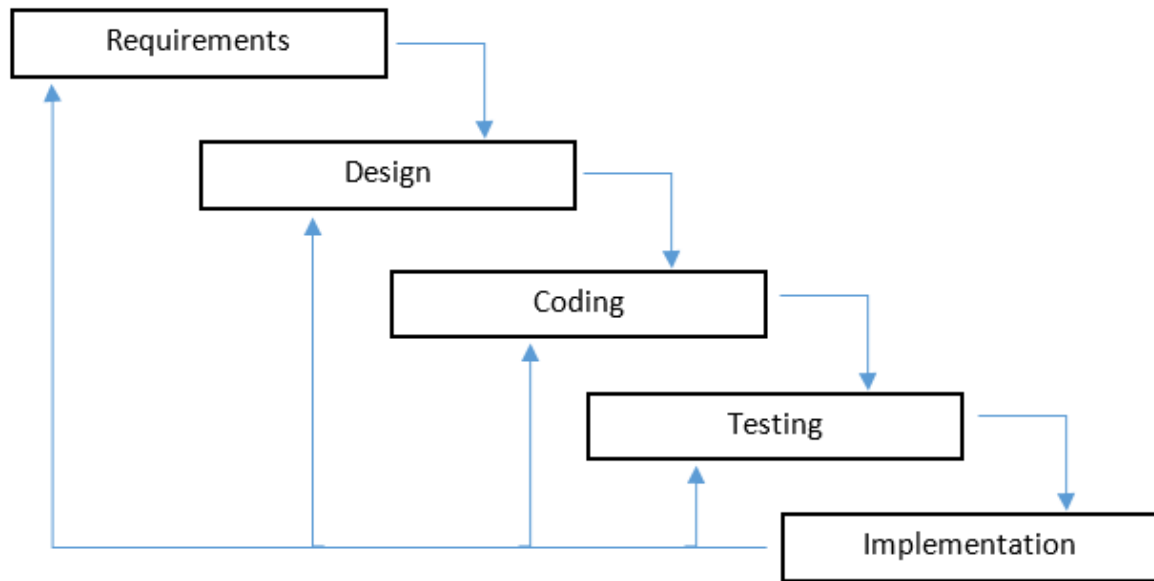


FIGURE 2.2 - Waterfall Model (Royce 1970)

According to Mohammed et al. (2010), the waterfall model was succeeded by the iterative development model. The study states that iterative development provides faster results to end users, with less documentation upfront and allows for more flexibility and feedback from users during the development process. Dawson (2009) summarises iterative development as collecting an initial set of requirements and then coding them. The iterative process follows a series of planned iterations as seen in Figure 2.3.

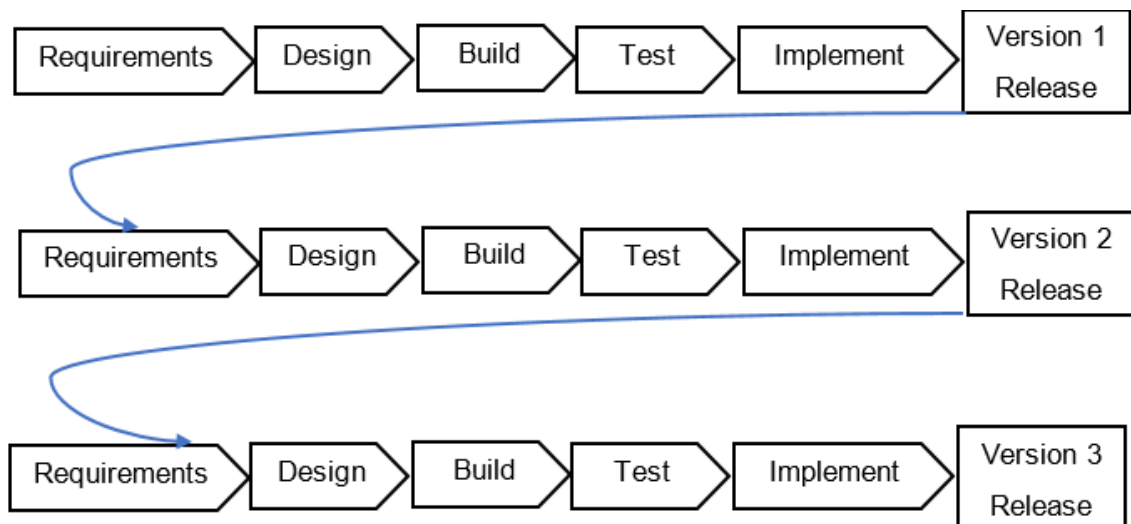


FIGURE 2.3 - Iterative Model (Dawson 2009)

In 2001, according to Highsmith et al. (2001) agile methods were created in response to the negative qualities of waterfall models. They state that agile is an incremental method



where the software is delivered as a series of functional areas. Dawson (2009) reveals that agile methods emphasise smaller development teams, face-to-face communication with the users and working software over documentation.

Keith et al. (2013) states that software development methodologies are either plan driven or agile based. Plan driven methods according to Keith et al. (2013) makes the assumption that most requirements can be specified in advance of design and development and mandates following a rigid linear approach to the SDLC. An example of a plan driven methodology is the waterfall model. Keith et al. (2013) argues that the waterfall approach is widely used for large complex software projects where requirements are known in advance of the design stage. According to Van Waardenburg & Van Vliet (2013) agile based planning depends less on a formal plan and more on ad hoc adjustments, based on feedback, throughout the development life cycle.

### **2.3 What are Agile Methods?**

In 2001, as described by Highsmith et al., (2001) the Agile Manifesto was signed by 15 software developers who recognised the need to change the way software was developed. They defined the following values of agile as:

- Individuals and interactions over process and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Highsmith (2002) states that agile methods address the need for speed and flexibility. He reveals that organisations today trade in an environment where change is constantly forcing software developers and organisations to be nimble, responsive and incorporate change on a daily basis. Furthermore, Highsmith (2002) states that the agile methodology is a collaborative environment in where development teams are encouraged to accept change by focusing on adaptability over a plan. Ceschi et al., (2005) define agile methods as a set of development techniques that put people at the centre of the software development process. According to them, comparing agile methods to waterfall or plan based methods; the use of agile methods delivers high-quality products faster and thereby, producing happier customers.

### **2.4 What is DSD?**

(Richardson et al. (2012) defines DSD as distributed projects involving two or more teams in different geographic locations. Holmström et al. (2006) describes the benefits of DSD

as proximity to markets, using virtual organisations to exploit market opportunities, access to skilled employees and lower labour costs. Persson et al. (2012) states that agile and DSD combined has emerged as a significant trend for organisations engaged in software development over the past ten years. In the early 2000's the desire for lower costs and access to a global resource pool were the main drivers for DSD. This remains the case today, but other factors are now accelerating the trend. According to Eppinger & Chitkara (2007) globalisation pressures have spawned a new paradigm where companies are utilising engineering skills dispersed around the world to develop products in a collaborative manner.

## **2.5 Benefits and challenges of agile methods and DSD**

According to Carmel & Agarwal (2001), DSD teams face temporal distance challenges and language barriers. Shrivastava & Rathod (2015) defines temporal distance as the difference in time zones whereby multiple teams work in different time zones thus making meetings difficult to plan. Language barriers according to Shrivastava & Rathod (2015) manifests themselves where DSD teams and sites do not share a common language. The study states that the impact of this is usually misinterpretation of requirements and poor information sharing between the DSD teams.

Ramesh et al. (2012) discuss the conflicting demands of DSD and agile methods. Their research reveals that agile methods will deliver the adaptability and the nimble approach that customers expect, but the rigor and process required by DSD is achieved by plan driven methodologies using traditional models like waterfall. According to Lehtinen et al. (2014) DSD and agile work on different principles, which makes the distributed agile projects difficult to manage. Their study states that DSD requires formal communication amongst team members while agile is based on informal communication with co-located teams working in close collaboration. Lehtinen et al. (2014) contends that several agile best practices including collaboration, face-to-face communication, self-organising teams and retrospectives, become more challenging with DSD.

Ramesh et al. (2012) explains that DSD relies on formal communication, control and shared documents. This is in conflict with one of the values of the agile manifesto where development teams respond to change over following a plan (Beck et al. 2001). A review of the literature on the topic suggests four dimensions are important when measuring the effectiveness of agile methods for DSD. These dimensions are as follows:

- Communication
- Process or Control
- Contracts

- Team cohesion

### 2.5.1 *Communication for Agile*

According to Ramesh et al. (2006) DSD relies on formal mechanisms like detail plans and software specifications to address and prevent communication issues that arise from geographic separation or lack of proximity. Beck et al. (2001) in the agile manifesto defines communication as individuals and interactions over process and tools. This creates a challenge with DSD teams in terms of balancing formal communications against the expected lack of formality with agile methods. Ramesh et al. (2006) in their research identified a number of ways to improve communications with DSD teams. They are as follows:

- Overlap of time zones and work hours with DSD teams
- Formalise the communications through a single point of contact for DSD teams
- Use project managers to coordinate activities between DSD teams
- Ensure constant communication using technologies like instant messaging

The study did not identify the degree to which agile evolved to a more plan driven or waterfall based model to facilitate the demands of the disturbed environment. For example, the introduction of a project manager to manage and coordinate development activities may be seen as regression to the waterfall model.

Abbott & Jones (2012) find that DSD is impacted by the geographic distance between client and the development team. They argue location does matter and that there can be advantages in being closer, geographically and temporally to clients. Kotlarsky & Oshri (2005) explains that the challenges in DSD are caused by time, cultural and geographical distances. With DSD teams, these geographic and temporal challenges play a large part in reducing the amount of face-to-face communication that naturally occurs with co-located teams. The research used a qualitative case study. The participant companies were two large multinational companies with geographically dispersed development teams. The research stated that successful collaboration for DSD is a combination of knowledge sharing, social ties and technology. However, the research does not explain if the use of agile methods is a contributing factor to successful collaboration.

Bavani (2012) argues that geographically distributed agile development teams face similar challenges as cited by Kotlarsky & Oshri (2005) in their study. Trust is key to distributed teams (Bavani 2012). The research states that one of the ways of achieving trust is by bringing everyone together face to face for release planning at the very beginning of the project. Bavani (2012) states that either you pay for travel costs at the beginning of the

project or continuously pay for defects during the project created through a lack of trust among the distributed teams.

Carmel & Agarwal (2001) points out that organisations can leverage technology for DSD teams. Examples of these technologies are email, voice mail, online discussion groups, project management tools, source code management tools, and bug tracking databases. These are asynchronous technologies and as such cannot be relied upon too much for communications. For example, a small issue regarding a bug can take days of back-and-forth over email to resolve, but a phone conversation can easily sort the problem. According to Carmel & Agarwal (2001) synchronous communication tools like telephone, audio conferencing, videoconferencing and desktop sharing are important. They play an important part in resolving miscommunications and misunderstandings over asynchronous technologies. The study reveals that some IT professionals shun video conferencing because of the awkwardness of interrupting colleagues. The study did not identify approaches or tactics to get IT professionals to use synchronous communications more effectively.

Carmel & Agarwal (2001) state that the quality of life is important for DSD teams. As such, there should be a minimum of four hours overlap between teams and time zones to facilitate good communication. The study discusses the goal of minimising time zone differences, thereby eliminating follow the sun type work that requires large time zone differences and further reduces effective communication.

### *2.5.2 Control Processes for Agile*

Ouchi (1978) noted that control is a central problem in hierarchical organisations because opportunities for miscommunication and distortion are so prevalent. The research established three types of controls that organisations use to manage towards objectives. Briefly, according to Harris et al. (2009) these control types are as follows:

- Behavioural control: Appropriate when the behaviours that transform inputs to outputs are known
- Outcome control: Appropriate when a process" output can be measured
- Clan control: Appropriate in ambiguous circumstances where neither the behaviours nor outputs can be predicted

Clan control is most closely aligned to agile methods and matches what Beck et al. (2001) describes as encouraging individuals and interactions over process and tools. Abrahamsson et al. (2002) explains that individuals and interactions are more important than rigour and process for software development projects. The research states the success of projects is more about the relationship and communality of developers

combined with the human role as opposed to rigorous processes. Ramesh et al. (2012), states that organisations rely on formal control processes to enact control for DSD teams, which contradicts the values of agile. The review of the literature shows some fundamental contradictions between values of agile software development and the use of DSD teams. Harris et al. (2009) propose that effective agile software development processes must provide clear control mechanisms to manage the progress and quality of the resulting software products.

Agile principles present control challenges related to balancing fixed vs. evolving quality requirements Ramesh et al. (2006). The study reveals that control over DSD teams is limited, and the DSD relies on fixed, upfront commitments on requirements. In contrast, agile development relies on negotiations between the DSD teams and the customer to determine if the requirement is complete to the customer's satisfaction. Whilst the study provides useful guidance on process controls mechanism to manage DSD developments it does not state if these controls were successful or provide a framework for implementing such controls.

Persson et al. (2012) conducted an in-depth case study of a successful agile distributed software project within two organisations. According to the research formal measurement and evaluation control together with agile's informal roles and relationships were successfully enacted through software collaboration tools. The findings of the research concludes that communication technologies can significantly support distributed agile practices by supporting and enabling both formal and informal project controls. For example, formal control uses a performance management strategy where outcomes are measured, evaluated and rewarded. Informal control measurement implies that norms, values or behaviours are implicitly specified and measured (Persson et al. 2012). Formal measurements are outcomes that are explicitly specified and measurable. The study states that formal rewards are bonuses and formal sanctions could be demotions. Informal rewards achieved through dialogue and discussion can be peer recognition, while informal sanctions could be social exclusion. The study finds that if applied in the right way, communication technologies can significantly support distributed, agile practices. However, the study used participants from only two organisations that spanned two time zones. Perhaps a study that covered three times zones with a time zone difference of more than 5 hours and more than two organisations would provide a better framework for implementing formal and informal controls for agile using DSD teams. Carmel & Agarwal (2001) and Wiredu (2011) both discuss the use of electronic meetings to make agile methods more effective. According to Wiredu (2011) the use of teleconferences and its most prominent characteristic of de-structured meetings support the use of agile for DSD

teams. Wiredu (2011) states that teleconferences offer software developers a medium for informal information sharing and a strategic opportunity for information processing for software process coordination.

Sarker & Sarker (2009) identify three dimensions of agility. These dimensions are process, people and linkage. The process dimension relates to the agility that is used in the team's systems development method guiding the project and its collaboration across time zones. People-based agility refers to the availability of skilled, flexible people at the different DSD team locations so that they can be deployed as and when necessary. Linkage agility pertains to the ability of the cultures of team members and stakeholders across locations to integrate. Perhaps the most interesting finding here is that distributed teams modified the agile development method. In some cases, the modified method became unworkable and had to be thrown out. The study found that agile is not effective because of the high degree of customisation of the agile methods for DSD, thereby turning it into waterfall. Other factors affecting the success of agile methods are lack of training provided to the DSD teams on agile methods and no perceived benefit to the development team of using agile methods. In fact, participants observed that DSD seriously affected their work-life balance with more demands on time and constant pressure.

Holmström et al. (2006) conducted a study featuring two companies using agile with globally DSD teams. The study identified time difference, geographic and cultural issues as barriers to DSD. The disadvantage of temporal distance is that the number of overlapping hours during a workday are reduced; team members have to be flexible to achieve overlap with remote colleagues. Holmström et al. (2006) identified no single agile method will solve the problems with globally DSD teams. Specifically, Holmström et al. (2006) found XP and Scrum methodologies were found to be particularly useful when working with distributed teams. The reason XP and Scrum are effective for DSD according to the research is, pair programming between DSD team members removed geographic distance and improved communication through shared ownership of the code across multiple locations. XP is an agile method that introduces the concept of pair programming (Holmström et al. 2006). Holmström et al. (2006) noted that time zone difference creates a feeling of being behind and missing out. The study also states that DSD teams changed their core working hours to create more overlap of time zones. However, the study did not discuss the impact of this on team morale, work-life balance and its possible negative effect on the effectiveness of agile methods. The second dimension or barrier to DSD identified by Holmström et al. (2006) is geographic distance. The problem here according to the study is how to create a feeling of team-ness among the distributed team members.

Holmström et al. (2006) finds that the Scrum planning practice creates a sense of togetherness. This is achieved using tools like post-it notes on a shared web page or virtual white boards that allow distributed teams to easily participate in the process. The third barrier to DSD identified by Holmström et al. (2006) is cultural differences. The study reveals that agile methods help increase mutual understanding and collaboration within and between teams. This study found that agile methods are suited to globally DSD. By using Scrum and XP methods, this study reveals that help the participant organisations to manage the complexity of the DSD projects.

Interestingly, Sommer et al. (2015) define an Agile/Stage-Gate Hybrid model combining a linear stage gate process with agile iteration cycle's. Stage gates are implemented at the project planning level and agile at the programming level. Stage gates use acceptance criteria to allow the project move from one stage to the next. However, one limitation of this study is that it does not address the DSD challenges using this Hybrid model.

### *2.5.3 Contracts for Agile*

According to Beck et al. (2001) agile encourages customer collaboration over contract negotiation. In contrast, Ivastava & Teo (2012) state that DSD is dependent on upfront targets, milestones, and a detailed set of requirements or in other words a contract. According to Ivastava & Teo (2012) a contract is key in facilitating interaction between DSD teams and its stakeholders.

Buslovič & Deribe (2012) reveals that daily stand-up meetings, contact with customers and other team members are challenged with DSD teams. According to the study, these challenges stem from the need for formal requirements documentation upfront and the imposition of contracts within DSD organisations. The study reveals that delivering business value as early as possible and reduces the risk of contract breach. The research states that significant collaboration required proper contracts. This contradicts one of the agile key values of customer collaboration over contract negotiation. However, the research does not offer any solution to building collaborative contracts for DSD teams using agile methods. According to Ceschi et al. (2005), companies using methods manage their customer relationships with flexible contracts instead of fixed contracts. Ceschi et al. (2005) does not address the bigger challenge of how companies regulate their relationships with their customers where organisations use agile methods for DSD.

### *2.5.4 Team cohesion for Agile*

Ramesh et al. (2012) states that team cohesion or a shared view of project goals are challenges associated with DSD. According (Holmström et al. 2006) to agile methods

combined with DSD exacerbates this challenge because it requires cooperation on all parts of the project. Carmel & Agarwal (2001) and Wiredu (2011) both discuss the predominance of electronic meetings to address these challenges. According to Wiredu (2011) the use of teleconferences and its most prominent characteristic of de-structured meetings support the use of Agile within globally DSD teams. Wiredu (2011) states that teleconferences offer software developers a medium for informal or organic information sharing and a strategic opportunity for information processing for software process coordination. The research conducted by Sharp & Ryan (2011) focusses on team structure to ascertain how organisations can be successful when using DSD teams. Sharp & Ryan (2011) argue that geographic and cultural differences do not affect agile methods succeeding in a distributed environment. The study cited time difference as the major challenge. Sharp & Ryan (2011) argue that overlap in working hours between DSD teams contributes to the success of agile for DSD.

## 2.6 Summary of Literature Review

Table 2.2 provides a list of literature reviewed for this research including the author, the year of publication, the research question and findings.

TABLE 2.2 - Literature reviewed for this study and findings

<b>Author</b>	<b>Year</b>	<b>Research Question</b>	<b>Findings</b>
Dr. Winston W. Royce	1970	Managing the development of large software systems	A seminal study by Royce that defined the waterfall model for delivering software systems
David E. Avison and Guy Fitzgerald	2003	Where Now for Development Methodologies	This paper explores the history of systems development methodologies, explores eras of development, and speculates on their future. The paper finds that we are in danger of returning to the bad old days of the pre-methodology because new methodologies lack control, standards and training.
B.W. Boehm	1998	A Spiral Model of Software	Boehm discusses the advantages and implications involved in using the Spiral



Author	Year	Research Question	Findings
		Development and Enhancement	model. He finds that: the spiral model is more adaptable to the full range of software project situations than are the primarily document-driven approaches such as the waterfall model.
Nabil Mohammed, Ali Munassar, A Govarddhan, Andhra Pradesh	2010	A Comparison Between Five Models Of Software Engineering	<p>This paper examines five development methods waterfall, Iteration, V-shaped, spiral and Extreme programming. These models have advantages and disadvantages. Therefore, the main objective of this research is to represent different models of software development and make a comparison between them to show the features and defects of each model. This research concluded that:</p> <ol style="list-style-type: none"> <li>1. There are many existing models for developing systems.</li> <li>2. These models were established between 1970 and 1999.</li> <li>3. The waterfall model and spiral model are used commonly in developing systems.</li> </ol>
Edward R. Davis, Alan M., Bersoff Edward H., Comer	1998	A strategy for comparing alternative software development life cycle models. Software Engineering	This paper provides a framework which can serve as a means to help software practitioners decide on an appropriate life cycle model to utilize on a particular project or in a particular application area.
Guus van Waardenburg,	2013	2013	Agile practices can coexist with plan-driven development keeping in mind the

Author	Year	Research Question	Findings
Hans van Vliet			context and take actions to mitigate the challenges incurred.
Mark Keith, Haluk Demirkan, and Michael Goul	2013	Service-Oriented Methodology for Systems Development	This aim of study was to implement a novel hybrid methodology based on concept of interdependence and coordination. The results imply that interdependence and coordination should be applied to the human processes involved in systems development in order to achieve better fit between project risk, interdependencies, and the selected methodology in order to improve overall project performance.
Martina Ceschi, Alberto Sillitti, and Giancarlo Succi	2005	Project management in plan-based and Agile companies	Requirements variables affect all the companies doing software development, however, agile based companies can better protect the customer from most of the negative effects
Balasubramaniam Ramesh, Kannan Mohan, Lan Cao,	2012	Ambidexterity in agile distributed development: An empirical investigation	This study conducted a multisite case study of three projects that use agile DSD to examine how these organizations pursue conflicting demands simultaneously. The findings, presented as a conceptual framework for dealing with the challenges posed by agile methods for DSD.
Julia Kotlarsky and Ilan Oshri	2005	Social ties, knowledge sharing and successful collaboration in globally distributed	The findings suggest that human-related issues, such as rapport and memory, are important for collaborative work in the teams studied. The paper concludes by discussing the implications for theory and suggesting a practical guide to enhance collaborative work in DSD

Author	Year	Research Question	Findings
		system development projects	teams.
Ita Richardson, Valentine Casey, Fergal McCaffery, John Burton, Sarah Beechama,	2012	A Process Framework for Global Software Engineering Teams	The study aim was to develop a global teaming process to address specific problems associated with DSD team. These problems are temporal, cultural, geographic and linguistic distance. The study found that that if managers are not implementing new global teaming practices they are putting their projects under threat of failure.
John Stouby Persson, Lars Mathiassen, & Ivan Aaen	2012	Agile DSD: enacting control through media and context	The study demonstrates that, if appropriately applied, communication technologies can significantly support distributed, agile practices.
Erran Carmel, Pamela Abbott	2001	Tactical approaches for alleviating distance in global software development	The study discussed some emerging approaches to get over the problem of distance in software development. Its findings discuss tactics for reducing:- distance, national and organizational cultural differences and temporal distance.
Suprika V. Shrivastava, Urvashi Rathod	2015	Categorization of risk factors for distributed agile projects	Analysis of qualitative data from interviews and project work documents resulted into categorization of forty-five DSD risk factors grouped under five core risk categories. The risk categories were mapped to Leavitt's model of organizational change for facilitating the implementation of results in real world.
Steven D. Eppinger and Anil R. Chitkara	2007	The new practice of global product	The study presents frameworks and perspectives to help in addressing the transformation to DSD and its

Author	Year	Research Question	Findings
		development	<p>implementation today. There is no blueprint, but organisations can more effectively plan for DSD by overcoming the following challenges:</p> <ol style="list-style-type: none"> <li>1.Management Engagement</li> <li>2.Process Modularity</li> <li>3.Product Modularity</li> <li>4.Core Development Skills</li> <li>5.Intellectual Property</li> <li>6.Data Quality</li> <li>7.Infrastructure</li> <li>8.Governance</li> <li>9.Collaboration</li> <li>10.Organisation Change Management</li> </ol>
<p>Helena Holmström, Brian Fitzgerald, Pär J. Ågerfalk, Eoin Ó. Conchúir</p>	2006	<p>Contemporary Practices in Systems Development Agile Practices Reduce Distance In Global Software Development</p>	<p>The study focusses on temporal, geographical and cultural distance its challenge to DSD practice. The problem according to the study is how to create a feeling of team-ness among the distributed team members. The study also found XP and Scrum methodologies are particularly useful with distributed team. XP is an agile method that introduces the concept of pair programming.</p>
<p>Pamela Y. Abbott Matthew R. Jones</p>	2012	<p>Everywhere and nowhere: Nearshore software development in the context of globalisation</p>	<p>The study finds that DSD is impacted by the geographic distance between client and the development team. They argue location does matter and that there can be advantages in being closer, geographically and temporally, to clients. They state that offshore development or DSD is a reality because of contemporary globalisation.</p>

Author	Year	Research Question	Findings
Raja Bavani	2012	Distributed agile, agile testing, and technical debt	The study argues that geographically distributed agile development teams find that trust is a key factor in determining its success. One of the ways of achieving trust is by bringing everyone together face to face for release planning at the very beginning of the project. Bavani states that either you pay for travel costs at the beginning of the project or continuously pay for defects during the project created through a lack of trust among the distributed teams.
William G. Ouchi	1978	The Transmission of Control Through Organizational Hierarchy.	Ouchi found that control is a central problem in hierarchical organisations because opportunities for miscommunication and distortion are so prevalent. The research established three types of controls that organisations use to manage towards objectives. They are behavioural, outcome and clan control.
Pekka Abrahamsson, Outi Salo & Jussi Ronkainen	2002	Agile software development methods	The study finds that individuals and interactions are more important than rigour and process for software development projects. The research states the success of DSD projects is about the relationship and communality of developers combined with the human role as opposed to institutional processes.
Gamel O. Wiredu	2011	Understanding the functions of teleconferences	The study discussed the predominance of electronic meetings to make agile methods more effective. According to

Author	Year	Research Question	Findings
		for coordinating global software development projects	study the use of teleconferences and its most prominent characteristic of de-structured meetings support the use of Agile within globally DSD teams. Wiredu states that teleconferences offer software developers a medium for informal information sharing and a strategic opportunity for information processing for software process coordination.
Saonee Sarker, Suprateek Sarker	2009	Exploring Agility in Distributed Information Systems Development Teams: An Interpretive Study in an Offshoring Context	The study reveals that agility should be viewed as three dimensions: resource, process, and linkage. Resource agility is based on the DSD team's access to necessary human and technological resources. Process agility pertains to the agility that originates in the team's systems development method enabling collaboration across time zones. Linkage agility arises from the nature of social relationships within the distributed team and with relevant project stakeholders, and is composed of cultural and communicative elements.
Rafael Prikladnicki and Jorge Luis Nicolas Audy	2012	Managing Global Software Engineering: A Comparative Analysis of Offshore Outsourcing and the Internal Offshoring of Software	The research compares offshore outsourcing and the internal offshoring of software development. The findings present the analysis of the differences in the challenges faced by companies and the patterns of evolution in the practice of software development in each business model.

Author	Year	Research Question	Findings
		Development	
Jason H. Sharp, Sherry D. Ryan	2011	Best Practices for Configuring Globally Distributed Agile Teams	This paper proposes a conceptual framework based upon the dimensions of agility, team structure, and virtualness to explore the question of how agile software development teams can be successfully configured in globally distributed environments. Based upon interviews conducted among five globally distributed agile teams a set of best practices is presented based upon these three dimensions.
SHIRISH C. SRIVASTVA,  THOMPSON S.H. TEO	2012	Contract Performance in Offshore Systems Development: Role of Control Mechanisms	Ivastava & Teo (2012) states that DSD is dependent on targets, milestones, and a detailed set of requirements or in other words a contract.
Timo O.A. Lehtinen,  Risto Virtanen,  Juha O. Viljanen,  Mika V. Mäntylä,  Casper Lassenius	2014	A tool supporting root cause analysis for synchronous retrospectives in distributed software teams	According to Lehtinen et al. (2014) DSD and agile work on different principles, which makes the distributed agile projects difficult to manage. The study states that DSD requires formal communication amongst the geographically distributed team members while agile is based on informal communication with co-located teams working in close collaboration.

## **2.7 Conclusion**

In summary and based on the review of the literature, organisations are increasingly using agile methods for DSD. As discussed and identified in the literature review there are numerous obstacles (see table 2.2 above) that need to be overcome if agile is to be effective for DSD. The literature review focussed on agile methods and DSD only. Findings from the literature review are that communication, process and control are necessary for agile to succeed with DSD. The principle of face-to-face communication is critical to the success of agile for DSD. According to the research, organisations are using software to enhance collaboration for DSD teams where face-to-face time is not possible due to time difference and geographic location. Furthermore, organisations are adjusting agile methods in a distributed setting to build more rigour and control into DSD projects.



### 3 METHODOLOGY AND FIELDWORK

#### 3.1 Introduction

This chapter describes the research approaches considered and appropriate to this research. The chapter starts with a brief overview of various research philosophies and research approaches. A discussion on the chosen methodology follows providing a rationale for choosing the methodology. It goes on to look at the limitations of the research and concludes with the ethics implementation and the lessons learned from choosing the research methodology.

#### 3.2 Research Philosophy

Saunders et al., (2009) defines the research process as an onion with six layers as illustrated in Figure 3.1. They suggest that the research philosophy chosen for any fieldwork will contain assumptions on how the researcher views the world.

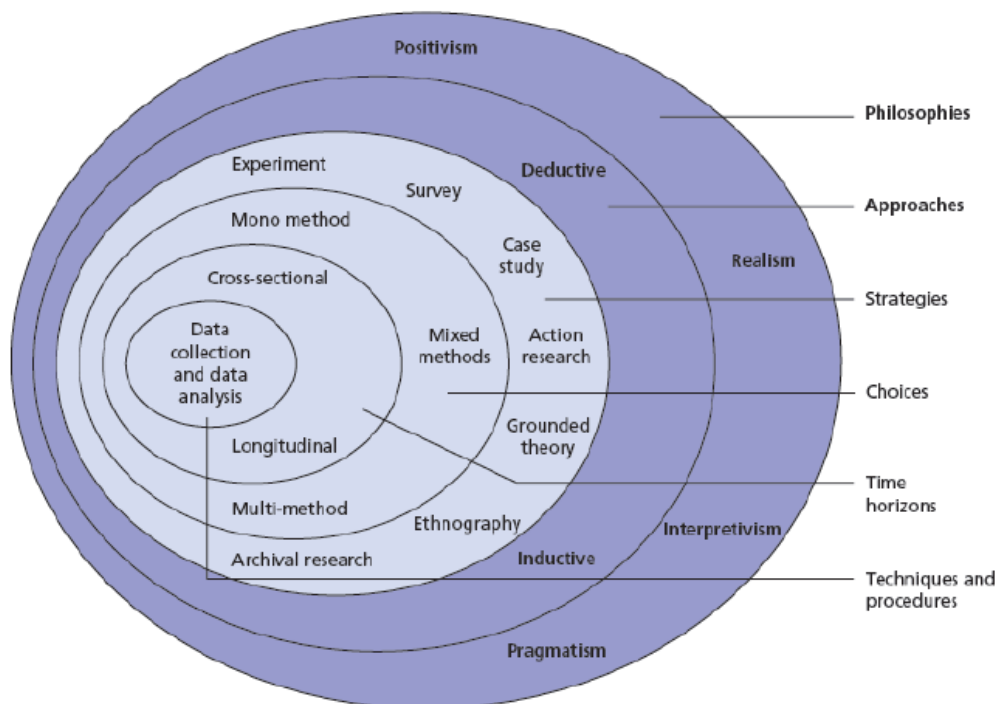


FIGURE 3.1 - Research Onion (Saunders, 2009)

Each layer is peeled back to reveal the research question at the centre of the research 'onion'. Before coming to the centre of the 'onion', Saunders et al., (2009) argues that there are important layers of the onion that need to be peeled away. The peeling away of these layers depicts the choices and challenges that need to be overcome before the researcher arrives at the research question. The six layers of the 'onion' are: research

philosophies, research approaches, research strategies, research choices, time horizons and eventually the method of collecting the data. The outer most layer of the research onion details the research philosophies available to researchers. According to Mingers (2001) the most common among them are: Positivism and Interpretivism. The research philosophies from Saunders et al. (2009) are outlined in Table 3.1.

TABLE 3.1 - Comparison of Four Research Philosophies Saunders et al. (2009)

	<b>Positivism</b>	<b>Realism</b>	<b>Interpretivism</b>	<b>Pragmatism</b>
<b>Ontology:</b> the researcher's view of the nature of reality or being	External, objective and independent of social actors	Is objective. Exists independently of human thoughts and beliefs or knowledge of their existence (realist), but is interpreted through social conditioning	Socially constructed, subjective, may change, multiple	External, multiple, view chosen to best enable answering of research question
<b>Epistemology:</b> the researcher's view regarding what constitutes acceptable knowledge	Only observable phenomena can provide credible data, facts. Focus on causality and law like generalisations, reducing phenomena to simplest elements	Observable phenomena provide credible data, facts. Insufficient data means inaccuracies in sensations (direct realism).	Subjective meanings and social phenomena. Focus upon the details of situation, a reality behind these details, subjective meanings motivating actions	Focus on practical applied research, integrating different perspectives to help interpret the data

	<b>Positivism</b>	<b>Realism</b>	<b>Interpretivism</b>	<b>Pragmatism</b>
<b>Axiology:</b> the researcher's view of the role of values in research	Research is undertaken in a value-free way, the researcher is independent of the data and maintains an objective stance	Research is value laden; the researcher is biased by worldviews, cultural experiences and upbringing. These will impact	Research is value bound, the researcher is part of what is being researched, cannot be separated and so will be subjective	Values play a large role in interpreting results, the researcher adopting both objective and subjective points

### **Positivism**

Mingers (2001) states that positivism provides a very superficial analysis when applied to very complex and political situations. The study states that from the positivist's perspective that the researcher should be detached from the entity they are studying. They state that a disinterested third party should complete the data gathering and analysis. Research conducted using a positivist philosophy uses a value free approach and the research is independent of the data collection process. According to Mingers (2001) positivism is very suited to statistical analysis or quantitative analysis. Saunders et al. (2009) describe how proponents of positivism lose rich insights by following a series of law like generalisations similar to those of the physical sciences. According to Trauth et al. (2000) the positivist philosophy assumes the data collection process has a well-defined and known task and is measured using a priori coding scheme setup in advance. Often, they state it does not capture interesting and important discussions that occur as part of the data collection process.

### **Interpretivism**

According to Trauth et al. (2000) by choosing interpretive methods, the researcher is acknowledging that access to the participants in the research comes through social constructions such as language, consciousness, and shared meanings. They state the interpretivist lends itself to inductively exploring human behaviour and communication in the context of the research study. As a result, there is no priori regarding the data to be collected. Trauth et al. (2000) states that relevant information emerges through an iterative process of examination and re-examination. They reveal that the interpretivist approach adopts an insider view of the participant's motivations and actions rather than

the outsider view of the positivist. Saunders et al. (2009) argues that an interpretive researcher is required to enter the social world of the research subjects and understand the subject's view of world.

### 3.2.1 Methodology chosen for this research

A detailed discussion on research philosophies is outside the scope of this work. This research leans more towards the interpretivism approach to research than it does to a positivistic approach because of the interactions between people, process, geographical/temporal distance and cultural influences. The study expects to gain additional insight from conducting an analysis from the interpretive paradigm.

## 3.3 Research Approach

According to Mingers (2012) there are three research approaches available. They are deductive, inductive and abductive. Mingers (2012) states that deduction is when particular instances are deduced to follow from general laws or assumed premises; induction, in which general laws are induced from particular examples or instances; abduction, where evidence is used to alter the probabilities associated with a particular hypothesis, that is, to confirm or disconfirm it.

### Deduction

Cooper & Schindler (2014) describe deduction as the process by which we test whether the hypothesis is capable of explaining the fact.

- The deductive approach involves the development of a theory and testing that theory through specifically designed research strategies. Figure 3.2 illustrates the deductive approach. Deductive reasoning works from the more general to the more specific and maybe called a top-down approach.

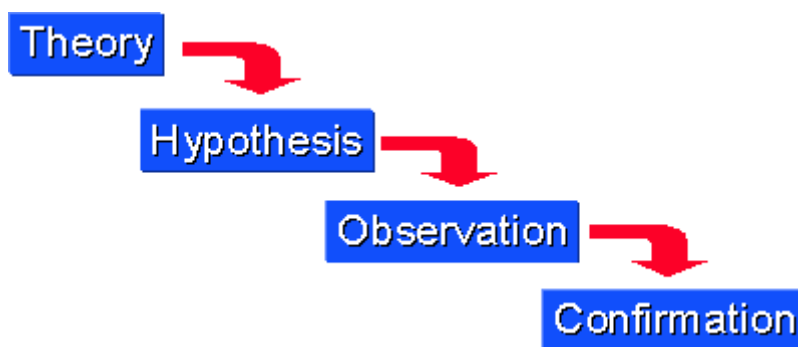


FIGURE 3.2 - Deductive Research Approach (Trochim 2006)

## Induction

Cooper & Schindler (2014) describe induction as occurring when we observe a fact and ask, "Why is this?" In answer to this question, we advance a tentative explanation (hypothesis). The hypothesis is plausible if it explains the event or condition (fact) that prompted the question. Figure 3.3 illustrates inductive research. The inductive approach moves from specific observations to broader generalisations and theories and is informally known as a bottom up approach.

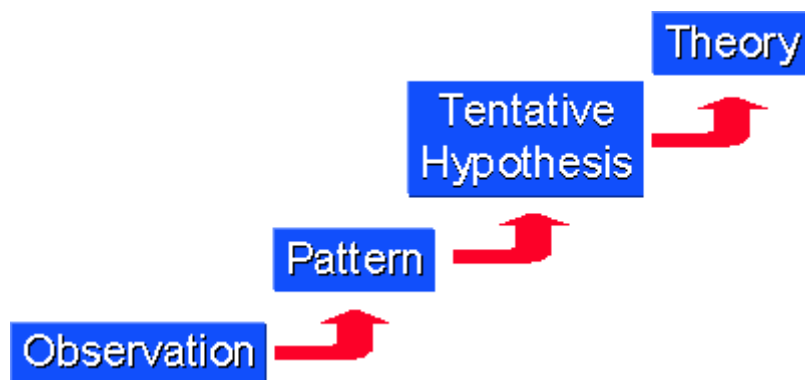


FIGURE 3.3 - Inductive Research Approach (Trochim 2006)

## Abduction

According to Mingers (2012) abduction begins with a particular occurrence or event, is unexpected, or does not conform to current theories. Following that, the researcher takes an imaginative leap to think of some theory or explanation, which might account for the event. Mingers (2012) describes this as exploratory hypothesis as to why the situation might have occurred.

### 3.3.1 Research Approach Chosen

This research is more exploratory and subjective in trying to understand the suitability of agile methods for DSDs teams.

Inductive research is more suited to this research project for the following reasons:

- It supports the collection of quantitative data using semi structured interviews
- Seeks an understanding of the meanings of social actors attached to events
- The researcher is very knowledgeable on Agile development methods and working with distributed teams
- Flexibility to allow changes of research emphasis as the research progresses
- Promotes the researcher as part of the research process

Deductive research involves the development of a theory and testing that theory through specifically designed research strategies. Based on these characteristics of deductive research, it is not a suitable approach for this research

### **3.4 Research Strategies Considered**

Cooper & Schindler (2014) describes experiments, surveys, case studies; interviews, action research, and grounded theory are common types of research strategies employed. Creswell (2003) states that during the 1990s the numbers and types of research strategies available to qualitative researchers became more clearly visible.

Creswell (2003) describes case studies as an approach, where the researcher explores in depth a program, an event, an activity, a process, or one or more individuals bounded by time and activity. According to Yin (2009) case study research is suitable for answering questions that start with how, who and why. Case studies are empirical investigations, in that they are based on knowledge and experience, or more practically speaking involve the collection and analysis of data. The survey strategy tends to be used in exploratory and descriptive research. It enables the collection of quantitative data and it is possible to generate findings that are representative of the whole population at less cost than collecting the data for the whole population. (Cooper & Schindler 2014).

Saunders et al. (2009) describes the experimental strategy as a form of research spawned by the natural sciences and used frequently in social science research. Experiments are common in exploratory and explanatory research to answer 'how' and 'why' questions. Experiments study causal links; assessing if a change in one independent variable produces a change in another dependent variable. According to Creswell (2003) the researcher tests a theory by specifying narrow hypotheses and collects data to support or repudiate the hypotheses. The simplest experiments are concerned with whether there is a link between two variables. The experimental strategy is useful for laboratory-based research, where research is undertaken in a controlled environment and is rigorous by nature.

According to Cooper & Schindler (2014) action research is designed to address complex practical problems about which little is known. For example, a scenario is studied; a corrective action is determined, planned and implemented.

Creswell (2003) describes grounded theory, whereby the researcher attempts to derive a general, abstract theory of a process, action, or interaction grounded in the views of participants in a study. This approach is suitable for quantitative studies. It generally involves multiple stages of data collection and refinement. The approach involves constant comparative analysis known as the Constant Comparative Method. The collected

data is organised into codes. The codes are grouped to develop concepts, and those concepts are then used to create categories from which a theory is created. With this approach the researcher is constantly comparing data with emerging categories and theoretical sampling of different groups to maximise the similarities and the differences of information.(Glaser & Straus 1967).

Semi-structured interviews face to face or by telephone are frequently used for qualitative research (Saunders et al. 2009). In advance of the interview, the researcher prepares a list of questions to be covered. The questions may vary from interview to interview and in fact, some questions are omitted depending on the interview situation. However, this can lead to difficulties during the data analysis stage. One of the benefits of semi-structured interviews is that, additional questions are available to investigate the research question.

#### *3.4.1 Research Strategy Chosen*

Action research is not suitable for this research study because of time constraints. After consideration this approach was deemed unsuitable for the scope of this dissertation owing to time required for data analysis, coding and constant revision. After assessing the experimental methodology as an approach for this research, it is not suitable, as it would not adequately answer the research question. A semi-structured interview is the best strategy to elicit the most valuable data taking into account the opinions, feelings and views of all stakeholders involved in distributed software teams using agile methods.

### **3.5 Research Choices**

Section 3.4 referred to quantitative and qualitative data as part of the discussion on research strategies. Quantitative methods are commonly used as a data collection technique that generates or uses numerical data Saunders et al. (2009). Conversely, Saunders et al. (2009) states that the qualitative technique is common for data collection or data analysis that generates or uses non-numerical data or text data. Researchers can choose between a single method (mono method), multiple or combined/mixed methods. A mixed method mode combines both qualitative and quantitative data collection techniques.

The preferred method of data collection for this dissertation is qualitative semi-structured interviews. This research choice for this research is a mono-method approach. According to Moore (1999) the advantages of semi-structured interviews are:

- The richness of the data collected

- Semi-structured interviews also allow informants the freedom to express their views in their own terms
- Allows the interviewer to ask more in depth questions during each interview
- Language use by participants was considered essential in gaining insight into their perceptions and values
- Contextual and relational aspects were seen as significant to understanding others' perceptions
- Data generated can be analysed in different ways

Semi-structured interviews have disadvantages as well:

- Interviewer bias – the interviewer may ask questions are biased based on his/hers experience.
- Availability of suitable participants – This study targeted senior IT staff within organisations in the U.K., Ireland and U.S. Arranging access to these participants, whose time is valuable, required a lot of preparation and organisation from the researcher. For example, three participant's interviews occurred at 10pm.

### **3.6 The Question Formulation Process**

A first draft of the proposed questions was shared with my supervisor in January 2015. Following feedback and discussion, the decision was made to run pilot interviews. In March 2015, two face to face pilot interviews were undertaken with colleagues who had experience of agile and DSD teams. The pilot was used to refine the questions, identify key constructs for the interviews and ensure that the interviews were no longer than sixty minutes. The pilot identified questions that were too leading and others that had more than one question that needed to be subdivided. The updated questions were submitted to the Ethics committee as part of the ethics application at the beginning of April 2015. Ethics approval was received by end of April 2015.

### **3.7 Data Collection**

The semi-structured interviews took place over a four-week period during the month of May 2015. Participants were selected for interview based on their experience with the research topic. Fifteen participants were contacted and provided with information about the research topic. Participants were asked to participate in a 60-minute face-to-face or Skype/telephone interview. Ten participants accepted requests for interviews and five participants either declined or did not respond.



Each participant was asked 15 questions. Participants were asked sub questions depending on the answer provided to the main question. The questions are available in Appendix 4.

Nine of the ten interviews were recorded with the permission of the participant. Permission to record one of the interviews was declined. Where permission to record was declined, the researcher took detailed notes for data analysis later. For all recorded interviews, the researcher took notes during the interview to create themes or categories as they arose during the interviews. These themes or categories were used during the data analysis stage to help with the coding or categorisation of data during the detailed analysis of each transcript.

### **3.8 Research Ethics**

Ethics documentation was submitted for approval on April 1<sup>st</sup> 2015. The ethics application consisted of the following details:

- School of Computer Science and Statistics Ethical Approval Form
- Participants Information Sheet for participants. This information sheet advised participants that the researcher was doing a Masters in Management of Information Systems in Trinity College Dublin. Participants were informed that the topic of the research was in relation to the effectiveness of agile methods with DSD teams.
- Participants Consent Form, which explained the background of the research, procedures of the study, intended publication and signed declaration. This consent form was signed prior to interview and sent via e-mail.
- Research Project Proposal application form. It stated the title of the project, purpose, description of methods used, participants involved, debriefing arrangements, ethical considerations and any relevant legislation.
- A document containing intended interview questions which included open and closed questions.

The application was reviewed by the Research Ethics Committee and the following revisions were requested:

- The participant information sheet should state that the researcher is looking for personal opinions and not the opinions of the organisation during interviews.
- A statement should be added to the declarations on the consent form saying that the participant is providing a personal opinion.
- Add a statement to the participant information sheet regarding the voluntary nature of participation.

- Add a statement to the participant information sheet that audio and not video will be recorded for interviews
- The following statement should be added to the interviews questions sheet :  
“Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to”.
- A statement should be added to the participant information sheet and consent form that participants right to withdraw at any stage “without penalty”
- The requested amendments were made and resubmitted, Ethics approval was granted to proceed on April 21st 2015.

### **3.9 Lessons Learned**

The following are the lessons learned during the course of the research study:

- Question formulation: Two pilot interviews were conducted. Because of these pilot interviews, the questions were revised to ensure the interview took no longer than 60 minutes.
- Interview Transcription: The first interview took almost six hours to transcribe. Due to the compressed timescales of the study, a transcription service was used to transcribe the remaining eight interviews. One participant-withheld permission for the interview to be recorded which resulted in detailed notes being taken during the course of the interview.
- Fifteen participants were targeted for interview but only ten participants took part. Researchers should ensure that they have a sufficient number of participants available once ethics approval is received to facilitate the collection of sufficient amounts of primary data.

### **3.10 Conclusion**

The data collection involved the execution of semi-structured interviews driven by the scope defined in Section 1.4. Participants were selected based on experience and seniority within their respective organisations. Participants were located in Ireland, U.S. and the U.K. Interviews were conducted on an iterative basis, allowing output from prior interviews to feed subsequent interviews based on emerging themes or topics.

## 4 FINDINGS AND ANALYSIS

### 4.1 Introduction

This chapter explains how the data was collected, analysed and provides a summary of the findings. Firstly, it looks at how the data that was collected for this research was analysed, namely using a data analysis software tool called Dedoose, and through employing the constant comparison method first explained by Glaser and Strauss (1967). The findings are then outlined and discussed in the latter part of this chapter. The chapter concludes with a summary of the research findings.

### 4.2 Participant Profile

Table 4.1 presents a profile of the participants interviewed for this research. The table maintains the anonymity of the participants but demonstrates their seniority within their respective organisations. It also shows the locations of their teams across multiple time zones. Seven of the participants are based in Ireland. One of the participants is based in the U.K. and two of the participants spend their time between India, U.S., U.K. and Ireland. Each of the participants have on average more than 8 years' experience specifically delivering DSD software projects.

TABLE 4.1 - Participant Profiles

Based in	Role	Years' experience of DSD	Distributed Locations	Participant Name
U.K.	Risk Manager	7	East and West Coast U.S., U.K., India, Hong Kong	Participant A
Ireland	V.P. Engineering	10	Central Europe, Ireland, India	Participant B
Ireland	Senior Developer	7	Dublin, U.K. and India	Participant C
Ireland	Programme Manager	10	Ireland, U.S., India and China	Participant D

Based in	Role	Years' experience of DSD	Distributed Locations	Participant Name
Ireland	Quality Assurance Lead	9	Ireland, Northern Ireland, East Coast U.S., China	Participant E
Ireland	Scrummaster	8	Eastern Europe, East Coast U.S., Dublin	Participant F
Ireland	Scrummaster	4	Ireland, East Coast U.S., Central Europe, India	Participant G
Ireland	Project Manager	8	India, U.S., Ireland	Participant H
U.K., U.S.	Derivatives Risk Project Manager	8	East Coast U.S., U.K., Central Europe, China	Participant I
India/Ireland	Managing Director	10	India, Ireland, West Coast U.S., U.K.	Participant J

### 4.3 Industrial Sector

The industrial sectors for each of the participants companies is shown in Figure 4.1. Participant A and participant I work in two of the biggest investment banks in London. Participant H works in one of the “Big Four” accounting firms. Participant D and participant G work for large multinational financial services companies. The remainder of the participants work in software development, telecommunications, and e-learning and insurance companies. A wide range of sectors are represented here which gives a good insight into the widespread use of DSD across sectors and helps to identify any differences in approach to DSD that there may be across sectors.

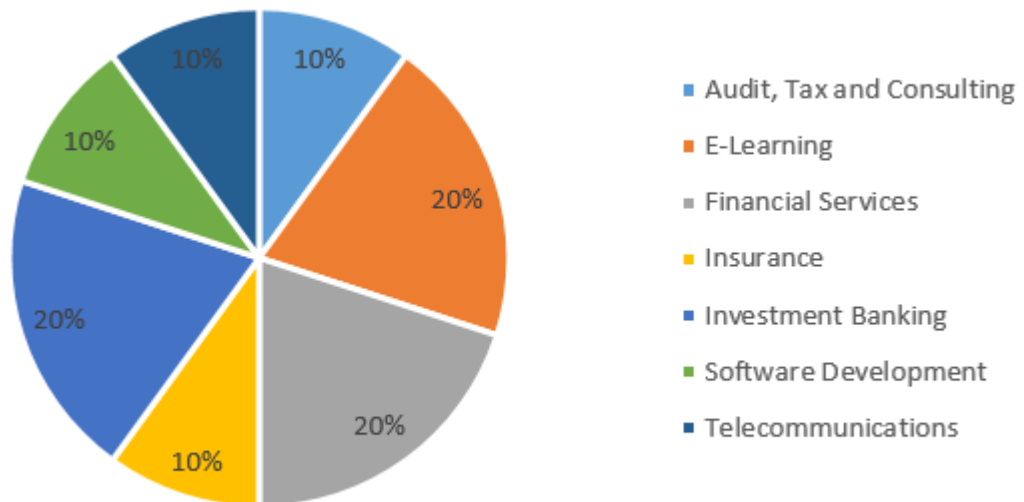


FIGURE 4.1 - Participants Industrial Sector

#### 4.4 Data Analysis

The method of data collection used for this research is qualitative semi-structured interviews. All interviews except one were recorded and all transcriptions were generated from audio recordings.

##### 4.4.1 Transcriptions

The researcher transcribed the first interview using Microsoft Word. This process took 6-8 hours to complete. Because of the amount of time required to complete a single transcription, the researcher used a third party transcription service for the remaining eight interviews. One interview was not recorded as permission to do so was withheld and detailed notes were taken during the interview process for use during data analysis. The outsourcing of the transcription service allowed the researcher to spend more time on the data analysis. A Computer Assisted Qualitative Data Analysis Software (CAQDAS) tool called Dedoose was used for the qualitative data analysis. Dedoose is designed by researchers to facilitate qualitative research. Dedoose ([www.dedoose.com](http://www.dedoose.com)) is used to build visualizations that expose patterns or themes in the data collected during the semi structured interviews.

##### 4.4.2 Codification

The transcripts were imported into Dedoose as shown in Figure 4.2.

Media					
Type	Title	Added	User	# Ex	Length
	Interview E.docx	06/06/2015	patm	9	46357
	Interview I.docx	06/06/2015	patm	25	68273
	Interview H.docx	06/06/2015	patm	26	71157
	Interview G.docx	06/06/2015	patm	19	26733
	Interview F.docx	06/06/2015	patm	34	42756
	Interview J.docx	06/06/2015	patm	10	49444
	Interview D.docx	06/12/2015	patm	10	36604
	Interview A.docx	06/06/2015	patm	23	43456
	Interview B.docx	06/06/2015	patm	23	94948
	Interview C.docx	06/06/2015	patm	15	56541

FIGURE 4.2 - Imported Transcripts

Each transcript was analysed and codes were created and assigned to keywords in the responses of each participant. These keywords are shown in Figure 4.3. Codes are labels that Dedoose uses to attach to keywords within the transcripts that relate to the research question. The coding system used is hierarchical and supports multiple parent/child levels. Figure 4.2 shows a hierarchical code structure as a result of applying codes to keywords in each of the transcripts. For example, the code 'Process/Control' has five subordinate levels as illustrated in Figure 4.3.



FIGURE 4.3 - Codification of Keywords within the Transcripts

The use of a CAQDAS named Dedoose provides functionality such as:

- Code Cloud

Code clouds are a visual representation of the codes applied to each of the transcripts. Code clouds are generated from the codes attached to keywords from the transcriptions. The code cloud gives greater prominence to keywords that appear more frequently in the transcription text as illustrated in Figure 4.4.



FIGURE 4.4 - Code Cloud taken from the Semi-Structured Interviews

Glaser & Straus' (1967) constant comparison method is used in this research to refine, merge and discard obsolete codes. Figure 4.5 shows the full list of codes applied to the transcripts before the process of refinement.



FIGURE 4.5 - Initial Codification of Transcripts.

- Code Co-Occurrence

The Code Co-Occurrence table provides information about how the code system is used across all transcription excerpts. The matrix in Figure 4.6 shows the frequencies for which all code pairings are applied to excerpts in the transcripts. It helps to expose both unexpected and expected patterns in the data where two codes are or are not used together. For example in Figure 4.6, three excerpts are coded with both the 'Process/Control' and 'Hybrid of Agile and Waterfall'. This indicates that as participants are thinking and responding on one of the questions, they discuss thoughts linked to other questions and topics. This type of matrix analysis demonstrates how participants naturally respond to key research questions.



Code Co-Occurrence														
Codes	Communication/Collaboration	Conflict Resolution	Lack of face to face	Video Conferencing	Management buy in	Time Zones	Process/Control	Hybrid of Agile and	More formal planning	More formal documentaton	Location of roles impacts	Synchronise work hours	Project Duration	Totals
Communication/Collaboration	1			1										1
Conflict Resolution		1												1
Lack of face to face		1	1				3		1					6
Video Conferencing	1		1	1	1	1	1	1			1		1	8
Management buy in				1	1	1	1	2			1		1	7
Time Zones				1	1	2	1	1			1		1	8
Process/Control			3	1	1	2	3	2			1	3	1	17
Hybrid of Agile and				1	2	1	3			1	1		1	10
More formal planning			1			1	2							4
More formal documentaton								1						1
Location of roles impacts				1	1	1	1	1					1	6
Synchronise work hours							3							3
Project Duration				1	1	1	1	1			1			6
Totals	1	1	6	8	7	8	17	10	4	1	6	3	6	

FIGURE 4.6 - Code Co-Occurrence

Table 4.2 provides an overview of codification in relation to each of the participants. The table is divided into a number of columns:

- Code – A code which has been identified based on the analysis of the transcripts
- Participant A to participant J – Each column identifies the number of times that participant referred to each of the codes identified
- Occurrence – The number of times the identified code was referred to by all participants
- No. Participants – The number of participants who referred to the identified code

TABLE 4.2 - Code Frequency

Code	Participant A	Participant B	Participant C	Participant D	Participant E	Participant F	Participant G	Participant H	Participant I	Participant J	Occurrences	No. Of Interviewees
<b>Communication/Collaboration</b>	0	0	0	0	0	1	1	0	0	0	2	2
Conflict Resolution	2	0	0	0	0	0	0	0	0	0	2	1
Lack of face to face communicatio	3	4	4	1	0	3	0	3	5	1	24	8
Video Conferencing	3	2	3	2	0	8	4	2	1	1	26	9
Sub Total	8	6	7	3	0	12	5	5	6	2	54	20
<b>Management Buy In</b>	3	0	2	1	0	5	3	5	2	1	22	8
Sub Total	3	0	2	1	0	5	3	5	2	1	22	8
<b>Time Zone Agility</b>	1	1	0	0	0	3	0	1	1	1	8	6
Sub Total	1	1	0	0	0	3	0	1	1	1	8	6
<b>Process/Control</b>	1	0	6	1	1	2	0	3	1	3	18	8
Hybrid of Agile and Waterfall	2	4	4	2	0	3	2	6	5	2	30	9
More formal planning needed	0	0	1	0	1	3	2	6	5	2	20	7
More formal documentation	1	1	0	1	0	0	0	0	2	0	5	4
Location of roles impacts success	0	0	0	2	0	3	0	0	1	1	7	4
Synchronise work hours	1	0	0	0	0	0	1	1	1	2	6	5
Sub Total	5	5	11	6	2	11	5	16	15	10	86	37
<b>Project Duration</b>	1	1	1	0	1	2	1	0	1	1	9	8
Sub Total	1	1	1	0	1	2	1	0	1	1	9	8
<b>Total</b>	17	12	21	10	3	30	14	26	24	14	0	171

An examination of codes was conducted firstly in descending order of frequency of use by participants and secondly by the frequency of occurrence across all participants. So for example, as illustrated in Table 4.2, 'Process/Control' and its sub codes are the top ranked codes. Therefore, the codification analysis identifies the top five themes in descending order of priority as Process/Control, Communication/Collaboration, and management buy in, project duration and time zone agility.

#### 4.5 Interview Findings

The main themes that emerged from all ten interviews, based on the codification within Dedoose using the Constant Comparative Method, are as follows:

1. Process/Control
  - a. Hybrid of Agile and Waterfall
  - b. Distribution of roles between DSD teams
  - c. Pure agile and DSD

2. Communication and Collaboration
3. Project Duration
4. Management Buy In
5. Time Zone Agility

It should be noted these themes have child levels or sub themes that are discussed in the findings as well.

#### 4.5.1 Process/Control

Process control relates to the application of agile methods within the DSD teams. All participants in this study are working with or had previous experience of working with agile methods and DSD teams. Participants were asked what agile methods they use to manage DSD. These questions were designed to uncover how agile is being adapted as a result of using DSD.

#### **Hybrid of Agile and Waterfall**

Interestingly, this study reveals that 50% of organisations as shown in Figure 4.6 are, in fact, using a *hybrid* of agile and waterfall to develop software using DSD teams. Organisations are modifying agile and introducing elements of waterfall to manage distributed teams. Participant A explains how his organisation uses a combination of agile and waterfall.

*“The initial project kick-off is a more traditional Waterfall approach. The software development team uses agile in the middle. Followed by a waterfall stage to deliver and implement the product.”* Quoted from Participant A<sup>1</sup>

Participant I, a senior project manager in an investment bank in London describes his modified agile approach as follows.

*“It's a mongrel. All of the projects that I have worked on or been involved in have never been pure agile textbook and they rarely have been waterfall either. They have been a mixture of the two.*

*The initial project kick-off or envisioning phase is a more traditional Waterfall approach followed by agile methods once software development commences.”*

*You have to have a mix of both, of both Waterfall and agile approach, to make sure that you succeed in delivering what you signed up to.”*

---

<sup>1</sup> See table 4.1, pp 34-35 for details on each participant.

Participant B, a software development manager with development teams in two geographic locations explains his approach to agile as a mixture of agile and waterfall called “wagile”.

*“It is not purely agile, it’s more wagile with more waterfall upfront followed by agile”*

In fact, he describes the agile and scrum evangelists as naïve where they implement pure agile for distributed teams.

Participant A describes how their organisation combines agile methods with the Six Sigma<sup>2</sup> framework. It works successfully where teams are co-located but less successfully for DSD team.

*“We combined agile and Six Sigma.... I have seen it working well in projects, again, where teams are co-located, but I haven’t so far seen successful delivery of projects where teams are dispersed using agile.”*

According to participant A, his organisation will continue to tailor the combined agile and Six Sigma approach until it succeeds with DSD.

Participant B talked about doing more design work upfront using the waterfall model and leaving the finer detail to the sprints.

*“So we try to do a certain amount up front, to make sure that we’re meeting the customer requirement and then when we get down to some of the finer details in the sprints”*

Participant H explains that the absence of a concept foundation phase up front affects the success of DSD. In this example, Participant H is benefitting from using waterfall upfront for the architecture and proof of concept phases and then follows that with agile methods for software development and delivery.

*“You may need a concept phase, for example, say, your project is 6 months long, that concept phase might be 2 weeks and out of that concept phase you would come out with some lightweight charter to say, this is the functionality, this is how much we think it’s going to cost, do we want to still proceed? Then you go into a foundation/architecture phase where you go down a little bit deeper, get some of the architecture questions out of the way.*

Interestingly, Participant I identified an upfront architecture design phase following the waterfall model to prepare the groundwork for agile methods.

---

<sup>2</sup> Six Sigma is a set of techniques and tools for process improvement. Motorola developed Six Sigma in 1986. Six Sigma seeks to improve the quality output of process by identifying and removing the causes of defects (errors) and minimizing variability in manufacturing and business processes.

*“It was a mistake looking back on it now not to use that more Waterfall as part of development to come up with the architectural diagram”*

From the interviews, a common theme that is emerging is that organisations are modifying agile methods and introducing elements of waterfall. Mostly, it seems that organisations are modifying agile at the requirements and design stage.

### **Pure Agile and DSD**

Participants from four organisations stated that agile methods are effective for DSD. These organisations use agile methods like Scrum to work with DSD teams. Participants C and F confirm that agile methods without modification or pure agile are successfully delivering in their respective organisations.

*“Without a doubt agile definitely gave us a higher percentage of success than hybrid or waterfall models”*

*“I’d say there is absolutely no problem for agile to really work in a DSD environment.”*

Interestingly, participant J explains that agile methods and DSD work for his organisation, In fact, he states using agile methods there is no need for a physical office for people to meet face to face. The face-to-face communication he says is easily achieved using technologies like Skype.

*“I could even go as far as saying that you could pull off a performing agile team without ever having to have an office where people meet.”*

### **Distribution of roles between distributed teams**

Participants were asked about the distribution of development team roles across time zones and geographic locations. The key roles in an agile team are product manager, project manager/scrum master, quality assurance, engineering and technical communication. The purpose of asking this question is to understand if there is an emerging best practice based on participants experience regarding what development team roles should be centralised and what roles should be remote. Participants stated that it is extremely important to the success of their projects to locate the business analyst or product owner close to the stakeholders or customers.

*“Just to be able to talk to the business and go off and find out information without having to have the problems of time zones”* Quoted from participant B.

*“My preference obviously is for the business analysts to be onshore”.* Quoted from participant D.

*“They’re leading those conversations back into the stakeholders that you’re delivering for”.* Quoted from participant I.

The reason for this is that the business analysts can easily discuss requirements without worrying about time zones or geographic separation. Agile methods emphasises customer collaboration and therefore, understanding and reacting quickly to changing customer requirements is important. The research indicates that this is the primary reason to locate the business analyst/product owner role close to the customer. Because of this lack of proximity, business requirements are shared with the DSD teams through formal documentation rather than informal communication. The use of formal documentation mitigates impediments of communications between the DSD teams. According to participant D, the business analyst should be centrally located and close to the customer.

*“Yeah, well the business analysts, my preference obviously is for the business analysts to be onshore. With agile require a greater onshore presence and closeness to the stakeholder group”*

Another interesting finding from the research is the role and physical location of the project manager. Participants reveal the importance of having the project manager co-located centrally with the business analyst. There are two reasons for this. The first is that projects are better managed centrally thus allowing organisations greater control and the ability to respond with agility to issues as they arise. Secondly, all decisions made on projects both strategic and tactical are made centrally with business analysts and senior management. The research indicates that having the project manager centrally located supports better control and communications. According to participant I,

*“The project manager was local because he namely, me, had got to be close to the business and to senior IT management... But when it comes to place like Pune, China, Moscow and then New York on the other side, we need to have the key people who can coordinate based in London.”*

Participant F states that the:

*“Scrummaster’s are based in Dublin, actually all our scrummasters are actually based in Dublin and developers in East Europe”*

The distribution of quality assurance or testing roles was mixed in terms of geographic dispersal. Fifty percent of the participant organisations had their quality assurance in a different time zone to development and business analysis. This is deliberate in that project managers want core people located close to key decision makers on the project. According to participant I

*“The QA lead was in London because we wanted to have core people in London”*

Participant B explains that they use a centralised QA lead or gate keeping role similar to a project manager. As a result, his organisation delivers higher quality software using a centrally managed quality assurance team that is geographically separate from the development team.

*“QA based in Dublin. It’s a safety net where we’re checking what’s coming back to make sure that the quality is correct”* Quoted from Participant B.

Another theme to emerge from the research is the importance of having a coordination role or a project manager to manage the delivery of the product from start to finish. Participant I gave an example of working on a project managed out of London. He described a scenario where the requirements are provided by the business analyst on East Coast of the US, developed in India, tested in China, and then returned to New York for sign off by the business analyst. The DSD team works across three time zones and any issues encountered in the delivery process require a project manager to manage the hand off or relationship between each time zone. For example, when engineers and business analysts have a difference of opinion over a requirement, the project manager needs to step in and resolve it, whilst at the same time ensuring the customer or stakeholders requirements are not impacted by hand over from one time zone to another.

#### *4.5.2 Communication and collaboration*

Participants were asked questions regarding their use of teleconferencing and videoconferencing to understand if/how these methods are replacing face-to-face communication. Participants were asked questions about how they carried out day-to-day communications both formally and informally. In addition, participants were asked to describe how they use technology between the different time zones where teams have a short window of time each day to share information.

All participants acknowledge that communication is much more difficult when working with DSD teams. The stand-up or scrum is an agile ceremony that takes place daily for 15 minutes. Regardless of location or time zone, the scrum must happen daily for agile to be effective. All participants are adamant that the only way to make DSD teams work is for all team members to attend stand-ups. This research indicates that the video conferencing tools like Skype, WebEx, Microsoft Lync, and Cisco Telepresence must be in place to foster successful communication. Ninety percent of participants stated that collaboration software, instant messaging and video conferencing are key to the effectiveness of agile. Participant G and participant H also state the technology is replacing face-to-face meeting. Participant G reveals that agile methods used in conjunction with Skype remove the need for face-to-face communication.

*“I could even go as far as saying that you could potentially pull off a performing agile team without ever having an office where people meet. You could just do it remotely and virtually”*

Interestingly according to participant H his organisation view the emergence of virtual office technology replacing face-to-face meetings. Participant H described his unusual experiences of using virtual office technology. His company uses this technology across three time zones. They use it to create multiple spaces or campuses based on geographic location. The software allows their teams to see who is in the office and whom they are talking to on a single screen or dashboard. It mimics the spontaneous communication of being co-located in a physical office.

*“DSD team members can actually see on their screen where everyone is, who is talking to who, join co-workers in a virtual room. Although I am working remote, I am still part of the team and I can see everybody just like I am physically in the office.”*

Just one participant mentioned email as their preferred mechanism of communication. Participant A identified technology as a barrier to communication. All video conferencing and voice calls are recorded. This prevents team members from speaking their minds as they felt the recordings could be used against them again.

*“People do not want to speak on the phone because they fear legal implications of their communications been recorded. That is quite challenging in DSD teams.”*

In fact, participant G stated that employees in his organisation are uncomfortable going on camera for video calls with DSD teams.

*“People didn’t feel comfortable with their cameras all the time.”*

In order to maintain good communication across time zones, participants in this research highlighted the importance of having communication tools readily available. One organisation has a designated room for video conferencing. Participant A described the situation in his organisation:

*“Availability of these rooms is limited and you can only get them at very limited times”.*

The room needed to be booked and in some cases was double booked. This problem was solved by using a designated team workspace or by providing each developer with the communication tools to easily join meetings.

Organisations must provide tools to maximize productivity as well. The research indicates how important the development and configuration management tools are to successfully executing agile with DSD teams. Collaborative development environments that support joined up working over the internet are mandatory. Sixty percent of participant



organisations use virtual storyboard software such as Trello or JIRA. Interestingly, one participant stated that staff attending meetings over the internet must have their camera's switched on. This is seen as a mechanism to keep staff engaged in meetings whilst simultaneously building trust with colleagues. Interview F described her experience of using video calls

*"Using a video, it's much more obvious if somebody is not involved and therefore somebody can pull them up on it. It is very easy to get distracted. Somebody tapped you on the shoulder when you are sitting at your desk, if you are on video it is much more, you are much more inclined to turn around and say, cannot talk now. Whereas without the video, yeah the temptation might be, particularly if it's somebody senior, to respond to them rather than be involved in the call"*

All participants stated they prefer to work with co-located teams. According to Prikladnicki & Audy (2012) one of the reasons organisations are engaging in distributed teams is to reduce the cost of delivering software. Two of the ten participants interviewed highlighted the hidden costs arising from managing and working with distributed teams. The agile manifesto as evangelised by Beck et al. (2001) recommends face to face conversations. Bringing the team together in one location for face-to-face meetings is key to project success. According to participant A, the cost of flying staff from London to India or vice-versa was one of the reasons these face-to-face meetings did not happen.

*"That person will be living here for three nights, staying overnight in hotel accommodation. All those, very, very quickly, add to your project costs"*

Lack of face-to-face communication is associated with hidden costs, such as, the time that project managers and business analysts spend writing up documents and talking on the phone to DSD teams. According to participant I

*"The hidden costs with DSD are the time that project managers and business analyst spend writing up documents and on the phone to people"*

#### 4.5.3 Management Buy In

Highsmith (2002) states that traditional project management methods are heavy on planning up front to build a contract with stakeholders/customers. These project-planning activities are not aligned to agile. According Beck et al. (2001), the Agile manifesto encourages customer collaboration over contract negotiation. Participants were asked if they used fixed price and fixed scope contracts for software development projects. Two of the participants refused to answer this question on the basis they were not privy to this information.

Participants from six organisations reveal that DSD teams are using agile methods for software development but project managers and senior management in these organisations follow a fixed price, fixed scope contract model. Participant C gave an example of what they do:

“I wouldn't say the our contracts are in agile focussed”

Management buy-in to agile is limited to the development teams:

Participant A stated that:

“Management look at Scrum and agile as purely an IT thing, where it's an organisation thing. That is actually a little bit of a struggle. You're not getting a buy-in across the organisation for Agile”

Participant I based on his experience stated that investment banking and agile methods with DSD teams are not compatible.

“You cannot deliver software for traders in investment banks and practice in an agile way with DSD teams because the traders just don't have time”

Participant F states that successful execution of DSD is linked to senior management support and knowledge of agile methods.

“We have strong advocates for agile at a senior level. Without that it wouldn't work”

#### 4.5.4 Project Duration

Participants were asked if shorter or longer duration projects were more suited to agile methods using DSD teams. Eighty percent of participants declared that a project with a duration of 3 months or more is suited to DSD and likely to succeed. Participants emphasised the importance of building up a stable team with knowledge of the business and technical platform. With shorter duration (less than 6 months) projects, the research revealed a co-located onshore setup would be best suited. Participant A stated that projects over three months are more suitable to agile for DSD.

*“Yeah, I think that it's better for the longer projects. Obviously, the challenges are there, but still it is manageable, whereas, for the shorter projects (3 months), the majority of the time the shorter projects are things that come up ad hoc, where people are looking for quick results, quick-win sort of solutions. Having distributed teams does not play, very well, in those scenarios”.*

The main reason according to this research is the time needed to get a team up to speed negates any benefits of using agile and DSD teams for projects with a duration of less than 3 months.

Participant G and participant F both stated that project duration does not affect the effectiveness of agile methods for DSD. Participant F stated the following:

*“Project duration does not have any impact”*

In addition, participant G stated

*“Our DSD teams work with short duration and long duration projects and it makes no difference to their effectiveness”.*

#### 4.5.5 Time Zone Agility

According to Beck et al. (2001) each day, the development team holds a meeting called the "daily scrum." The scrum or stand-up should be held in the same location and at the same time each day. The challenge for this research is to understand how stand-up works for DSD teams in India, U.S. and Ireland. These scrum meetings are strictly time-boxed to 15 minutes to keep the discussion short but relevant. The daily scrum meeting is not used as a problem solving or issue resolution meeting. Issues that are raised are taken offline and usually dealt with by the relevant staff immediately after the meeting. Each team member (both local and distributed) is asked the same three questions each day:

- What did you do yesterday?
- What will you do today?
- Are there any impediments in your way?

Participants were asked how they manage work across multiple time zones and effectively communicate across multiple time zones. This research reveals the challenge of coordinating teams across multiple time zones. Sixty percent of participants stated that simple tasks like scheduling a meeting between team members across multiple times in Ireland, India and U.S. proved difficult. In particular, when unplanned communication is needed to discuss an issue the challenge of operating across multiple time zones proves very difficult in terms of getting team members to attend a teleconference or videoconference.

According to participant E simple tasks like scheduling stand-up requires planning.

*“The guys in China & India report directly to us in Dublin. The guys in the US are east coast based so you’re looking at five hours max, we generally have a stand up at half one GMT. That’s the kind of closest time that everyone can come together”*

Participant G reveals that good coordination skills are essential to facilitate communication between DSD teams. But, in addition to that, he reveals that Ireland is uniquely advantaged in that it can manage overlap with India and China in the morning and overlap with the U.S. the afternoon.

*“Working 9:00 a.m. to 5:00 p.m. on U.S. Eastern Time, versus Indian time, you have a very short window of overlap between the two. You need to coordinate well. That is why it is good to be in a place like Ireland. It’s in the middle of these time zones.”*

While this approach is beneficial to information sharing and communication, the impact on the work-life balance needs to be taken into account. According to participant G

*“In Asia and also in the East Coast or West Coast of the U.S., they used to work late in the evenings or early mornings, to make up for all those meetings”*

Participant I stated that using agile for DSD is not effective with four time zones. He gave an example of a scenario with a business analyst and developers in New York, a project manager in London, testing teams in India and China.

*“The major problem is getting all the team members together on a daily Scrum. If you take for example, like my current project, we have testers in China, testers in India, management in London and one business analyst and a developer in New York. It is almost impossible to get everyone together at the same time without somebody having to make a major sacrifice. We can only do that at most once a week”.*

In order to attend the daily Scrum, team members would need to be innovative and change their working hours. Again, the sustainability of this model is questionable as work-life balance is compromised.

The agility of co-located teams is taken for granted where face to face communications and unplanned meetings are part of the culture and happen organically. The findings from this research suggests that having project managers who plan based on the different environments and time zones is crucial. A number of participants cited the fact that agile and DSD requires much more planning when it comes to organising meetings and sharing information. Participant E described the example of working with a team in India, China and Dublin. The project is managed out of Dublin, with India and China doing the development and testing respectively. The challenge in this case is that there is a short overlap between time zones to schedule scrums and other formal and informal communications with remote teams. According to participant E

*“You have a very short window of overlap between the time zones. You need to coordinate well.”*

*“We have to look at all the time zones and say, okay when can we have our regular scrum times, when can we have our regular review times. That’s a lot more difficult.”* Quoted from participant F.

#### 4.6 Summary of Findings

All participants were asked how effective agile is within their organisation for DSD. The response from each of the participant companies is shown in Figure 4.7. Of the 10 participants 50% of the respondents said it is effective but as a hybrid using elements of Waterfall. Four respondents were certain that pure agile is effective. One respondent was adamant that Waterfall works best for DSD because of its focus on upfront planning and formal documentation.

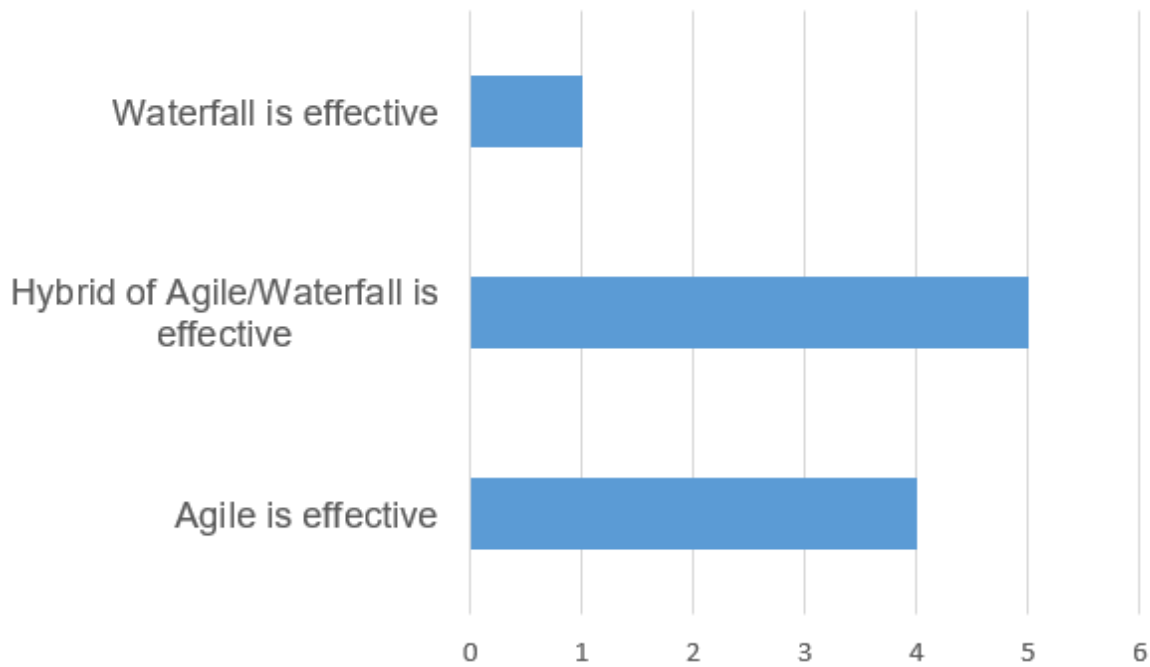


FIGURE 4.7 - Effectiveness of Agile Methods

Table 4.3 provides an overview of the findings in relation to the themes identified at the beginning of this chapter in Section 4.5.

TABLE 4.3 - Summary of Research Findings

Themes	Findings	No of citations
Process/ Control	Hybrid of agile and waterfall to develop software	
	Agile is effective for co-located teams	10
	Wagile/Mongrel Methodology - Waterfall upfront, agile in the middle followed by waterfall again for implementation	5
	Six Sigma/Agile methodologies combination	1

<b>Themes</b>	<b>Findings</b>	<b>No of citations</b>
	Waterfall upfront for proof of concept stage followed by agile for the remainder of the project	5
	Schedule face to face project kick off meeting at head office and invite DSD teams to attend	6
	Agile methods are effective for DSD	4
<b>Process /Control</b>	<b><i>Distribution of DSD team roles</i></b>	
	The business analyst must be close to customer	10
	The project manager should be located centrally with the business analyst. The reason for this is closeness to strategic decision makers and enabling better coordination and control of the DSD milestones.	8
	Project Manager required for issue resolution across DSD sites. The project manager needs to specifically manage the hand off or relationship between each DSD team	8
	Embed trusted developers from head office with the DSD team. This also increases the cost of DSD teams.	4
	Quality assurance staff are not co-located with DSD teams by design.	5
<b>Communication/ Collaboration</b>	DSD teams and lack of face to face communication increase costs because more documentation is required	2
	Without Skype agile and DSD will not work.	9
	Agile methods used in conjunction with Skype removed the need for face-to-face communication.	2
	Organisations see technology replacing face to face meetings	6

<b>Themes</b>	<b>Findings</b>	<b>No of citations</b>
	Technology is a barrier to communication. All video and voice calls are recorded. This prevented team members from speaking their minds as they felt it could be used against them again	1
	Technology is as a barrier to communication. All video conferencing and voice calls are recorded. This prevented team members from speaking their minds as they felt it could be used against them again	1
	All DSD teams' members must attend meetings with their cameras on. This ensured DSD team members engage in the meeting rather than multi-tasking. It also facilitates virtual face-to-face time with remote colleagues.	4
	All participants are adamant that the only way to make DSD teams work is for all team members to attend stand-ups	10
	Lack of face to face communication increases the hidden costs such as the time that project managers and business analysts spend writing up more formal requirements documents and talking on the phone/video conference to DSD team members	5
<b>Management buy in</b>	Investment banking, DSD, and agile methods are not compatible.	2
	Successful execution of DSD is linked to senior management support and knowledge of agile methods	8
<b>Project Duration</b>	The time to get a team up to speed negates any benefits of using agile and DSD teams for projects with a duration less than three months.	6
<b>Time zone agility</b>	Simple tasks like scheduling a meeting between team members, traversing multiple times in Ireland, India and U.S. proved difficult	8

Themes	Findings	No of citations
	Good coordination skills are required to facilitate communicate between DSD teams. However, in addition to that, Ireland is uniquely advantaged in that it can manage overlap with India and China in the morning and overlap with the U.S. the afternoon.	5
	In order to attend the daily Scrum, team members need to be innovative and change their working hours. Again, the sustainability of this model is questionable as work-life balance is severely compromised	5
	Using agile methods for DSD is not effective with four time zones due to difficulty coordinating the various teams and finding a suitable time that overlaps	6

Fifty percent of the participants revealed they use waterfall initially to collect requirements, use agile in the middle to develop the software product and then return to waterfall for the beta testing and implementation of the software. Based on the findings, this research introduces the Hybrid DSD lifecycle as shown in Figure 4.8. This research reveals that DSD organisations are spending more time producing requirements and design documents using a waterfall approach. The development process then follows an agile approach for software development and reverts to waterfall for testing and implementation. The extensive literature review undertaken for this study was unable to find a similar Hybrid DSD lifecycle introduced by previous research. A similar study by Sommer et al. (2015) introduces an Agile/Stage-Gate Hybrid model combining a linear stage gate process with agile iteration cycle's. However, there are a number of differences regarding between the Hybrid DSD lifecycle put forward by this research and the Agile/Stage-Gate Hybrid. For example, the Hybrid DSD lifecycle uses the waterfall model for requirements gathering, design, beta testing and implementation stages. The Agile/Stage-Gate Hybrid from Sommer et al. (2015) using stage gates for project management and strategic decisions only. They manage requirements, design, beta testing and implementation stages using agile methods. In addition and according to the study Agile/Stage-Gate is suitable for companies using co-located development teams.



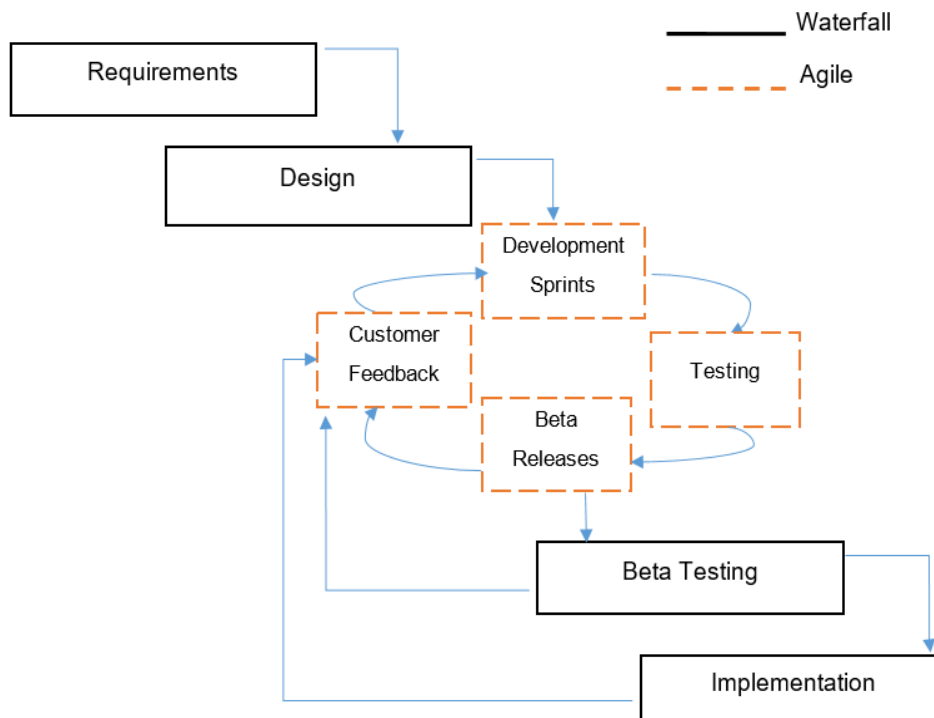


FIGURE 4.8 - Hybrid DSD Lifecycle

Additionally the research introduces a guidance framework for organisations to assist with the implementation and operation of Hybrid DSD methods. Table 4.4 provides a set of heuristics or DSD Implementation Guidance Framework based on the research findings to provide further guidance to organisations for using hybrid methods for DSD.

TABLE 4.4 - DSD Implementation Framework

Themes	Findings	Framework for DSD
<b>Process/Control</b>	Hybrid of agile and waterfall to develop software	Document and establish an agreed development method for DSD teams. This shall include <ul style="list-style-type: none"> <li>Choose a development method for DSD that works</li> <li>Provide training and guidance for development methods</li> <li>Accept change - The DSD development method will evolve based on feedback from the</li> </ul>

Themes	Findings	Framework for DSD
		teams
<b>Process/Control</b>	Distribution of DSD team roles	<p>Establish a DSD role and responsibilities strategy. This shall include:</p> <ul style="list-style-type: none"> <li>• Breakdown of roles between DSD locations</li> <li>• Locate project Management function close to key decision makers</li> <li>• Locate the business analyst close to the customer</li> <li>• Locate scrummaster with development team</li> <li>• Locate QA lead close to the project manager and business analyst</li> <li>• Embed trusted developers from head office with the DSD teams</li> </ul>
<b>Communication/ Collaboration</b>		<p>Establish a DSD collaboration strategy for communicating between DSD team locations. The strategy shall include:</p> <ul style="list-style-type: none"> <li>• Provide technology to support virtual face to face meetings</li> <li>• Train DSD teams members on using the technology</li> <li>• Ensure DSD team members are comfortable on video calls</li> <li>• Mandate the use of video to promote virtual face to face meetings</li> </ul>
<b>Management buy in</b>		<p>Establish a senior management/stakeholder engagement and communication process. This process shall include:</p> <ul style="list-style-type: none"> <li>• Senior management representation and attendance at weekly or monthly status meetings.</li> </ul>
<b>Time zone</b>		<p>Implement a communication strategy to manage task across time zones:</p> <ul style="list-style-type: none"> <li>• Identify skills needed for communication and</li> </ul>

Themes	Findings	Framework for DSD
		<p>planning within DSD teams</p> <ul style="list-style-type: none"><li>• Establish a time where all DSD teams can attend to facilitate stand-ups and information sharing meetings</li><li>• Plan meetings well in advance</li><li>• Project managers should have excellent coordination skills</li><li>• Encourage formal and informal communication with DSD teams</li><li>• Establish a format requirements gathering process that is subject to change control</li></ul>

## 5 CONCLUSIONS AND FUTURE WORK

### 5.1 Introduction

This chapter discusses how the findings of the study answer the main and sub research questions. Also discussed in this chapter is the contribution of this research to the existing body of knowledge, the limitations of the research and possible recommendations for future research in the area.

### 5.2 Answering the research questions

The primary research question for this study is:

*How effective are Agile Methods for Distributed Software Development?*

Based on data collected from ten semi-structured interviews, the answer to this question is “*Agile methods are at times ineffective for DSD*”. The research shows that for some scenarios agile methods are ineffective for DSD whereas in others they seem to be more effective. While all participants agree that agile works effectively with development teams co-located, the majority (60%) of the participants state that agile methods, when combined with DSD teams, are ineffective. As a result, this research reveals that organisations are modifying agile methods by introducing elements of waterfall to support agile when used with DSD.

Participants reveal, as shown in Figure 5.1, that they use waterfall initially to collect the product requirements, agile in the middle to develop the software product and then return to waterfall for the testing and implementation of the software. For example, business requirements documents, typically used with the waterfall model, are preferred instead of using user stories, typically used in agile development, for requirements gathering. This research reveals that DSD organisations are spending more time documenting requirements and building design documents using the waterfall model. Once these are complete, agile is introduced. This process is represented in Figure 5.1 as the iterative development, testing, incremental release and customer feedback stages of the Hybrid DSD lifecycle introduced by this research in section 4.6. The final two stages are beta testing and implementation and they follow a waterfall approach.

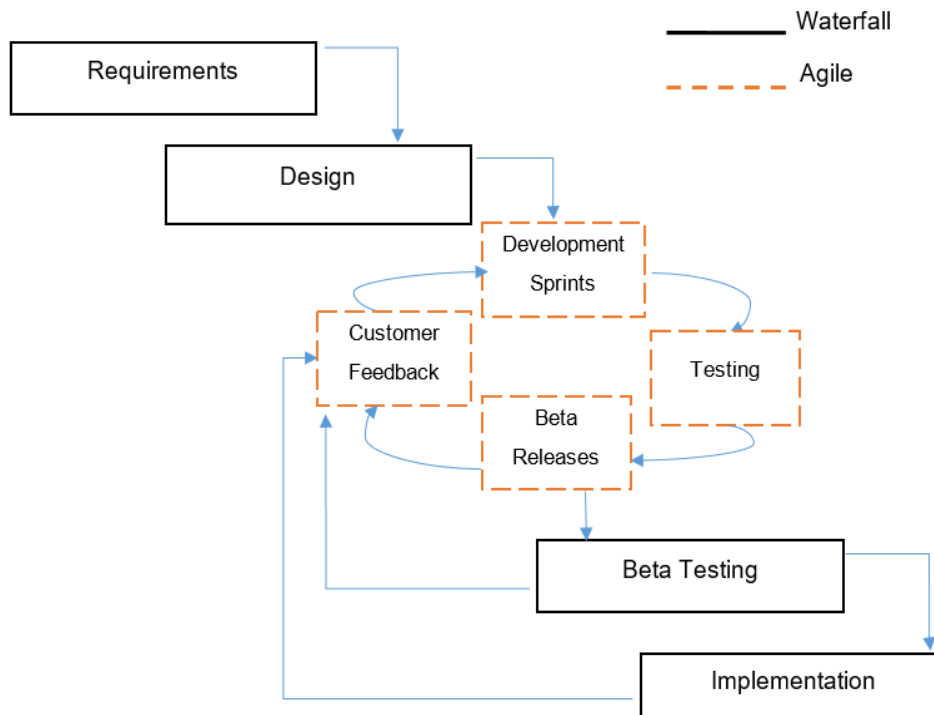


FIGURE 5.1 - Hybrid DSD Lifecycle

As explained earlier in the findings section 4.5, one DSD organisation stated that they mandate the use of video for all developers attending meetings remotely. The reason for doing this is to encourage face-to-face communication over video and build trust in the same way as meeting face to face. Therefore, this new finding suggests that organisations using DSD need policies in place for the operation and management of meetings over video. Another interesting finding from this research is that agile methods for DSD are not effective with four or more time zones due to difficulty coordinating disparate teams and finding a suitable time that suits all DSD teams.

An important sub-question posed in this research was:

*What factors influence the choice of using waterfall based methods over agile methods in distributed software development?*

The answer to this question is much less clear. Participants were asked this question to help the researcher better understand the advantages of using the agile methods over the waterfall model in a DSD environment. The majority (60%) of participants are using a hybrid of agile and waterfall to effectively manage the DSD teams. All participants stated that using agile methods with co-located teams works effectively. Therefore, the

introduction of DSD teams instead of co-located teams has to be considered as the reason organisations are using a hybrid of agile and waterfall models.

Management buy-in is another factor that influences organisations to choose waterfall over agile. The findings reveal that a number of organisations use agile development methods but the customer contracts are waterfall based. In fact, these contracts have payment milestones that follow a waterfall model and contradict the agile method of delivery. An interesting finding of this research is that customer contracts must take agile methods in account when DSD teams and agile methods are used. According to participant G

*“We have agile at the development team’s level and it’s working well. Then your middle-layer managers are asking you for Gantt charts and business requirement documents linked to payment milestones, which is really a waterfall style approach. The managers higher up the chain, they need to understand that it’s a different concept for dealing with customers and our contracts should reflect that”.*

The next sub-question is:

*What factors make specific projects more suited to agile development in distributed software development than others?*

This research question considers what projects are more suited to agile methods for DSD. The question also aims to identify project preconditions for DSD. The research emphasises the importance of building up a stable team with knowledge of the business and technical platform. With shorter duration (2-3 months) projects, the research reveals a co-located onshore setup would be best suited. Participant E stated

*“Smaller projects with 2-3 sprints over a 3 month period are best suited to a co-located team ... with distributed teams you could spend most of your time getting up to speed”*

The research revealed the projects with a duration of 3 months or more are best suited to agile and DSD.

The third sub question is:

*Is there a framework that can help organisations better manage distributed software development?*

Based on the findings from this research, a simplistic framework is presented; see Figure 5.1(or section 4.6). The function of the framework is to better guide organisations in the successful use of the agile methodology for DSD. This research has taken a first step in

identifying a dedicated framework for the use of agile methods in DSD. Further research will be required to further develop and validate this framework for other DSD organisations across diverse industries, each with their own individual challenges.

### **5.3 Limitations of the Research**

The sample size was small and is therefore not fully representative of all senior managers working with agile and DSD teams. Given, the extensive experience of over eighty years in total of the participants, the researcher is confident that while not all views maybe represented in this research the majority is likely to be present. It is worth noting that a limitation in this study is the purely qualitative nature of the study.

### **5.4 Recommendations for Future Work**

During the course of this research and in particular at the data analysis stage, the following questions and topics were uncovered as areas for future research.

#### *5.4.1 A case study on the effectiveness of a Hybrid DSD Lifecycle*

Based on the findings, this research has introduced a Hybrid DSD lifecycle. Future research in this area could explore the full potential of the Hybrid DSD lifecycle to deliver for organisations developing software with multiple geographic locations.

#### *5.4.2 A broader study*

The sample size of this study is relatively small using semi structure interviews. A larger qualitative survey using semi-structured interviews run by organisations like Ambler (2015) would be interesting in order to get a more representative view of senior managers and IT leaders who use agile methods for DSD. Perhaps as a more efficient method for gathering data from larger sample sizes, a questionnaire could instead be given to these individuals who are often time poor.

#### *5.4.3 Are banking organisations suited to agile methods for DSD?*

The financial services sector is heavily regulated with a reliance on complicated back office systems. Traditionally, this sector has used waterfall models with its predictable predefined outcomes.

## **5.5 Conclusion**

This research set out to investigate the effectiveness of agile methods for DSD. The motivation for doing this research is twofold; firstly, the researchers background of over twenty years working in the software development industry, secondly his experience of the challenges of working with agile methods for DSD. A comprehensive literature review was undertaken to position the research question in the context of the literature review. The data was collected for analysis using qualitative semi-structured interviews. The overall findings present considerable number of contradictions between agile methods and their benefits for DSD. As a result, the findings were used to create a Hybrid lifecycle better suited for DSD.

The Hybrid DSD lifecycle is a first step in identifying a dedicated framework for DSD and further research will be required to validate the framework. To sum up, the software development industry is continually evolving. We have seen the evolution of development methods over the past fifty years driven by customer expectations and needs. The current delivery methods of agile combined with DSD are not perfect as the research has shown and will require further academic research to provide advancements and frameworks that will benefit commercial industry.



## 6 References

- Abbott, P.P. & Jones, M.R., 2012. Everywhere and nowhere: Nearshore software development in the context of globalisation. *Computer and Information Science*, 21(5), pp.529–551. Available at: <http://dx.doi.org/10.1057/ejis.2012.7>.
- Abrahamsson, P. et al., 2002. Agile Software Development Methods: Review and Analysis. *VTT Publications* 478, 478(3), pp.167–168. Available at: <http://eljabiri1.tripod.com/sitebuildercontent/sitebuilderfiles/Comp-agile.pdf>  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.161.5931&rep=rep1&type=pdf>.
- Ambler, S., 2015. Scott W. Ambler's Home Page. Available at: <http://www.amblysoft.com/scottAmbler.html> [Accessed November 19, 2014].
- Bavani, R., 2012. Distributed agile, agile testing, and technical debt. *IEEE Software*, 29, pp.28–33.
- Beck, K. et al., 2001. Manifesto for agile software development. Available at: <http://www.agilemanifesto.org/>.
- Beck, K. et al., 2001. Manifesto for Agile Software Development. *Agile Alliance*, pp.1–2. Available at: <http://agilemanifesto.org>.
- Boehm, B.W., 1988. A Spiral Model of Software Development and Enhancement. *Computer*, 21(5), pp.61–72.
- Buslovič, M. & Deribe, S., 2012. A Multiple Case Study on Contradictions and Pre-conditions for Outsourcing Agile Software Development Projects.
- Carmel, E. & Agarwal, R., 2001. Tactical approaches for alleviating distance in global software development. *IEEE Software*, 18(April), pp.22–29.
- Ceschi, M. et al., 2005. Project management in plan-based and Agile companies. *IEEE Software*, 22(June), pp.21–27.
- Cohen, D., Lindvall, M. & Costa, P., 2004. An Introduction to Agile Methods. *Advances in Computers*, 62(C), pp.1–66.
- Conchúir, E.Ó., Ågerfalk, P.J. & Olsson, H.H., 2001. Global Software Development : Where are the Benefits ? , pp.127–132.
- Cooper, D.R. & Schindler, P.S., 2014. *Business research methods*, McGraw-Hill/Irwin.
- Creswell, J.W., 2003. Creswell, J.W. (2003). Chapter One, "A Framework for Design." *Research design Qualitative quantitative and mixed methods approaches*, pp.3–26.
- Dawson, C.W., 2009. *Projects in Computing and Information Systems*, Available at: [http://www.sentimentaltoday.net/National\\_Academy\\_Press/0321263553.Addison.Wesley.P](http://www.sentimentaltoday.net/National_Academy_Press/0321263553.Addison.Wesley.P)

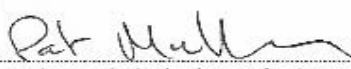
- ublishing.Company.Projects.in.Computing.and.Information.Systems.A.Students.Guide.Jun.2005.pdf.
- Eppinger, S.D. & Chitkara, a. R., 2007. The new practice of global product development. *IEEE Engineering Management Review*, 35(1).
- Fitzgerald, G. & Avison, D.E., 2003. Where Now for Development Methodologies. *Communications Of The ACM*, pp.78–82.
- Glaser, B.G. & Straus, A., 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*,
- Harris, M.L., Hevner, A.R. & Collins, R.W., 2009. Controls in Flexible Software Development Controls in Flexible Software Development. *Communications of the Association for Information Systems*, 24(June), pp.757–776.
- Highsmith, J., 2002. *Agile Software Development Ecosystems*,
- Highsmith, J., Consortium, C. & Cockburn, A., 2001. Development : The Business of Innovation. *Science*, 34, pp.120–123. Available at: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=947100](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=947100).
- Holmström, H. et al., 2006. CONTEMPORARY PRACTICES IN SYSTEMS DEVELOPMENT AGILE PRACTICES REDUCE DISTANCE IN GLOBAL SOFTWARE DEVELOPMENT. *Information Systems Management*, pp.7–19.
- Holmström, H. et al., 2006. Agile Practices Reduce Distance in Global Software Development. *Information Systems Management*, 23(3), pp.7–18.
- Ivastava, S.C. & Teo, T.S.H., 2012. Contract Performance in Offshore Systems Development: Role of Control Mechanisms. *Journal of Management Information Systems*, 29(1), pp.115–158. Available at: <http://mesharpe.metapress.com/openurl.asp?genre=article&id=doi:10.2753/MIS0742-1222290104> [Accessed November 18, 2014].
- Keith, M., Demirkan, H. & Goul, M., 2013. Service-Oriented Methodology for Systems Development. *Journal of Management Information Systems*, 30(1), pp.227–260. Available at: <http://mesharpe.metapress.com/openurl.asp?genre=article&id=doi:10.2753/MIS0742-1222300107>.
- Kotlarsky, J. & Oshri, I., 2005. Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *EUROPEAN JOURNAL OF INFORMATION SYSTEMS*, 14, pp.37–48.
- Lehtinen, T.O. a et al., 2014. A tool supporting root cause analysis for synchronous retrospectives in distributed software teams. *Information and Software Technology*, 56(4), pp.408–437. Available at: <http://dx.doi.org/10.1016/j.infsof.2014.01.004>.
- Mingers, J., 2012. Abduction: the missing link between deduction and induction. A comment on Ormerod's "rational inference: deductive, inductive and probabilistic thinking." *Journal of*

- the Operational Research Society*, 63(6), pp.860–861. Available at: <http://dx.doi.org/10.1057/jors.2011.85>.
- Mingers, J., 2001. Combining IS Research Methods : Towards a Pluralist Methodology. *Information Systems Research*, 12(3), pp.240–260.
- Mohammed, N. et al., 2010. A Comparison Between Five Models Of Software Engineering. *IJCSI International Journal of Computer Science Issues*, 7(5), pp.94–101.
- Moore, N., 1999. *How to Do Research: The Complete Guide to Designing and Managing Research Projects*,
- Ouchi, W.G., 1978. The Transmission of Control Through Organizational Hierarchy. *Academy of Management Journal*, 21(2), pp.173–192.
- Persson, J.S., Mathiassen, L. & Aaen, I., 2012. Agile distributed software development: Enacting control through media and context. *Information Systems Journal*, 22, pp.411–433.
- Prikladnicki, R. & Audy, J.L.N., 2012. Managing Global Software Engineering: A Comparative Analysis of Offshore Outsourcing and the Internal Offshoring of Software Development. *Information Systems Management*, 29(3), pp.216–232. Available at: <http://www.tandfonline.com/doi/abs/10.1080/10580530.2012.687313> [Accessed November 18, 2014].
- Ramesh, B. et al., 2006. Can distributed software development be agile? *Communications of the ACM*, 49(10), pp.41–46. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-33749583819&partnerID=40>.
- Ramesh, B., Mohan, K. & Cao, L., 2012a. Ambidexterity in Agile Distributed Development : An Ambidexterity in Agile Distributed Development : An Empirical Investigation. *Information Systems Research*, 23(January 2015), pp.323–339.
- Ramesh, B., Mohan, K. & Cao, L., 2012b. Ambidexterity in agile distributed development: An empirical investigation. *Information Systems Research*, 23(2), pp.323–339.
- Richardson, I. et al., 2012. A process framework for global software engineering teams. *Information and Software Technology*, 54(11), pp.1175–1191. Available at: <http://dx.doi.org/10.1016/j.infsof.2012.05.002>.
- Royce, W.W., 1970. Managing the Development of Large Software Systems. *IEEE WESCON (Reprinted in Proceedings Ninth International Conference on Software Engineering.)*, (August), pp.1–9.
- Sarker, S. & Sarker, S., 2009. Exploring Agility in Distributed Information Systems Development Teams: An Interpretive Study in an Offshoring Context. *Information Systems Research*, 20(3), pp.440–461. Available at: <http://pubsonline.informs.org/doi/abs/10.1287/isre.1090.0241> [Accessed November 18, 2014].
- Saunders, M., Lewis, P. & Thornhill, A., 2009. *Research Methods for Business Students Fifth edition*,

- Sharp, J. & Ryan, S., 2011. Best Practices for Configuring Globally Distributed Agile Teams. *Journal of Information Technology Management*, XXII(4). Available at: <http://jitm.ubalt.edu/XXII-4/article4.pdf>.
- Shrivastava, S. V. & Rathod, U., 2015. Categorization of risk factors for distributed agile projects. *Information and Software Technology*, 58, pp.373–387. Available at: <http://www.sciencedirect.com/science/article/pii/S095058491400161X> [Accessed December 1, 2014].
- Sommer, A.F. et al., 2015. Improved Product Development Performance through Agile/Stage-Gate Hybrids: The Next-Generation Stage-Gate Process? *Research-Technology Management*, 58(1), pp.34–45. Available at: <http://openurl.ingenta.com/content/xref?genre=article&issn=0895-6308&volume=58&issue=1&spage=34>.
- Trauth, E.M., Hall, H. & Jessup, L.M., 2000. UNDERSTANDING COMPUTER-MEDIATED DISCUSSIONS POSITIVIST AND INTERPRETIVE ANALYSES OF GROUP SUPPORT SYSTEM USE. *MIS Quarterly*, 24(1), pp.43–79.
- Trochim, W.M.K., 2006. Deduction & Induction. Available at: <http://www.socialresearchmethods.net/kb/dedind.php> [Accessed August 16, 2015].
- Vijayaraghavan, P.K., Sundararajan, S. & Bhasi, M., 2014. Case study on risk management practice in large offshore-outsourced Agile software projects. *IET Software*, 8(6), pp.245–257. Available at: <http://digital-library.theiet.org/content/journals/10.1049/iet-sen.2013.0190>.
- Van Waardenburg, G. & Van Vliet, H., 2013. When agile meets the enterprise. *Information and Software Technology*, 55(12), pp.2154–2171. Available at: <http://dx.doi.org/10.1016/j.infsof.2013.07.012>.
- Wiredu, G.O., 2011. Understanding the functions of teleconferences for coordinating global software development projects. *Information Systems Journal*, 21, pp.175–194.
- Yin, R.K., 2009. CASE STUDY RESEARCH Design and Methods Third Edition. *Applied Social Research Methods Series*, 5.

## Appendices

### Appendix 1 – Ethics Application

<b>School of Computer Science and Statistics Research Ethical Application Form</b>		
<b>Part A</b>		
Project Title: <b>How Effective are Agile Methods for Distributed Software Development?</b>		
Name of Lead Researcher (student in case of project work): <b>Pat Mulhern</b>		
Name of Supervisor: <b>Maria Maloney</b>		
PUID E-mail: <b>mulhernp@tcd.ie</b>	Contact Tel No.: <b>0872393720</b>	
Course Name and Code (if applicable): <b>MSc in Management of Information Systems</b>		
Estimated start date of survey/research: <b>1st May 2015</b>		
I confirm that I will (where relevant):		
<ul style="list-style-type: none"> <li>• Familiarize myself with the Data Protection Act and the College Good Research Practice guidelines <a href="http://www.tcd.ie/info_compliance/dp/legislation.php">http://www.tcd.ie/info_compliance/dp/legislation.php</a>;</li> <li>• Tell participants that any recordings, e.g. audio/video/photographs, will not be identifiable unless prior written permission has been given. I will obtain permission for specific reuse (in papers, talks, etc.)</li> <li>• Provide participants with an information sheet (or web-page for web-based experiments) that describes the main procedures (a copy of the information sheet must be included with this application)</li> <li>• Obtain informed consent for participation (a copy of the informed consent form must be included with this application)</li> <li>• Should the research be observational, ask participants for their consent to be observed</li> <li>• Tell participants that their participation is voluntary</li> <li>• Tell participants that they may withdraw at any time and for any reason without penalty</li> <li>• Give participants the option of omitting questions they do not wish to answer if a questionnaire is used</li> <li>• Tell participants that their data will be treated with full confidentiality and that, if published, it will not be identified as theirs</li> <li>• On request, debrief participants at the end of their participation (i.e. give them a brief explanation of the study)</li> <li>• Verify that participants are 18 years or older and competent to supply consent.</li> <li>• If the study involves participants viewing video displays then I will verify that they understand that if they or anyone in their family has a history of epilepsy then the participant is proceeding at their own risk</li> <li>• Declare any potential conflict of interest to participants.</li> <li>• Inform participants that in the extremely unlikely event that illicit activity is reported to me during the study I will be obliged to report it to appropriate authorities.</li> <li>• Act in accordance with the information provided (i.e. if I tell participants I will not do something, then I will not do it).</li> </ul>		
Signed: 	Date: <b>1/4/2015</b>	
Lead Researcher/student in case of project work		
<b>Part B</b>		
<i>Please answer the following questions.</i>		
Has this research application or any application of a similar nature connected to this research project been refused ethical approval by another review committee of the College (or at the institutions of any collaborators)?	<i>No</i>	
Will your project involve photographing participants or electronic audio or video recordings?	<i>Yes</i>	
Will your project deliberately involve misleading participants in any way?	<i>No</i>	
Does this study contain commercially sensitive material?	<i>No</i>	
Is there a risk of participants experiencing either physical or psychological distress or discomfort? If yes, give details on a separate sheet and state what you will tell them to do if they should experience any such problems (e.g. who they can contact for help).	<i>No</i>	
Does your study involve any of the following?	Children (under 18 years of age)	<i>No</i>
	People with intellectual or communication difficulties	<i>No</i>

Patients

**School of Computer Science and Statistics  
Research Ethical Application Form**

**Part C**

I confirm that the materials I have submitted provided a complete and accurate account of the research I propose to conduct in this context, including my assessment of the ethical ramifications.

Signed: Pat Mallon Date: 11/4/2015  
Lead Researcher/student in case of project work

*There is an obligation on the lead researcher to bring to the attention of the SCSS Research Ethics Committee any issues with ethical implications not clearly covered above.*

**Part D**

If external or other TCD Ethics Committee approval has been received, please complete below.

External TCD ethical approval has been received and no further ethical approval is required from the School's Research Ethical Committee. I have attached a copy of the external ethical approval for the School's Research Unit.

Signed: \_\_\_\_\_ Date: \_\_\_\_\_  
Lead Researcher/student in case of project work

**Part E**

If the research is proposed by an undergraduate or postgraduate student, please have the below section completed.

I confirm, as an academic supervisor of this proposed research that the documents at hand are complete (i.e. each item on the submission checklist is accounted for) and are in a form that is suitable for review by the SCSS Research Ethics Committee.

Signed: Marie Molloy Date: 30/03/2015  
Supervisor

Completed application forms together with supporting documentation should be submitted electronically to [research-ethics@scss.tcd.ie](mailto:research-ethics@scss.tcd.ie). Please use TCD e-mail addresses only. When your application has been reviewed and approved by the Ethics committee hardcopies with original signatures should be submitted to the School of Computer Science & Statistics, Room 137, O'Reilly Institute, Trinity College, Dublin

## Appendix 2 – Information Page for Participants

# TRINITY COLLEGE DUBLIN

## INFORMATION SHEET FOR PARTICIPANTS

**Researcher:** Pat Mulhern, School of Computer Science and Statistics, Trinity College Dublin.

**Contact Details:** mulherp@tcd.ie

This dissertation is to be submitted to the School of Computer Science and Statistics of Trinity College Dublin in partial fulfilment of the requirements for the degree of Masters of Science in Management of Information Systems.

**Background of Research:** This research attempts to investigate the effectiveness of the agile development methodologies for organisations using globally distributed software development teams. However, there are obstacles that need to be overcome if agile is to be effective for distributed software development. Not all projects can be delivered using distributed agile methods. For large and complex projects the traditional approach of using the waterfall methods may in fact be a better choice than agile. Research indicates that organisations are modifying agile methods in a distributed setting to build more rigour and control into their software development process. Other challenges associated with agile and distributed development are the lack of face to face communications, time difference and geographic location. As part of this research, the goal is identify and propose solutions to these challenges associated with agile and distributed teams.

**Procedures of this study:** This research will be conducted via semi structured interviews involving senior managers, software developers, product managers, project managers and quality assurance staff working in software development companies.

**Declarations of conflicts of interest:** I intend to interview my colleagues for the purpose of this study. Interview participants will come from organisations within the software development industry that have experience of, or have worked in globally distributed software development teams. I can declare that no conflicts of interest have been identified for the proposed research.

**Participation:** You were selected to participate because you work in software development sector. As such you can make a valuable contribution regarding research in the area of globally distributed agile development teams and their effectiveness.

**Participants Time:** The time taken to participate in the interview will be no more than 40-60 minutes. Participation is voluntary and you may withdraw at any time and for any reason without penalty. You also have the right to omit individual responses without penalty. On request, participants will be debriefed at the end of their participation and provided with a copy of the research.

**Interview Structure:** The interview will consist of a series of questions relating to the workings of globally distributed agile development teams. With permission from the participant, the interview will be recorded on an audio recording device for transcription and analysis by the researcher. The participant may opt out of the recording at any time.

No recordings will be made available to anyone other than the researcher and the recordings will not be replayed in any public presentation of research. Only audio will be recorded for interviews conducted over video conferencing.

**Confidentiality:** The anonymity of the participant will be preserved in analysis, publication and presentation of resulting data and findings. In the extremely unlikely event that illicit activity is reported I will be obliged to report it to appropriate authorities. The researcher is looking for the participants' personal opinions and not the opinions of the participants' organisation.



### Appendix 3 – Informed Consent Form

# TRINITY COLLEGE DUBLIN INFORMATION CONSENT FORM

**Researcher:** Pat Mulhern, School of Computer Science and Statistics, Trinity College Dublin.

**Background of Research:** This research attempts to investigate the effectiveness of the agile development methodologies for organisations using globally distributed software development teams. However, there are obstacles that need to be overcome if agile is to be effective for distributed software development. Not all projects can be delivered using distributed agile methods. For large and complex projects the traditional approach of using the waterfall approach may in fact be a better choice than agile. Research indicates that organizations are modifying agile methods in a distributed setting to build more rigour and control into their software development process. Other challenges associated with agile and distributed development are the lack of face to face communications, time difference and geographic location. As part of this research, the goal is to identify and propose solutions to these challenges associated with agile and distributed teams.

In addition to the primary research question I have set the following research objectives:

1. To identify the effectiveness of distributed software development across 2 or more countries or geographic locations
2. To identify the factors that influence the choice of using waterfall based methods over agile methods
3. To identify factors that make specific projects more suited to agile development than others
4. To identify a framework that can help organizations better manage distributed development projects

**Procedures:** This research will be conducted via semi structured interviews involving senior managers, software developers, product managers, project managers and quality assurance staff working software development companies.

**Publication:** This dissertation is to be submitted to the School of Computer Science and Statistics of Trinity College Dublin in partial fulfillment of the requirements for the degree of Masters of Science in Management of Information Systems. Individual results will be aggregated anonymously and research reported on aggregate results.

#### **DECLARATION:**

- I am 18 years or older and am competent to provide consent.
- I have read, or had read to me, a document providing information about this research and this consent form. I have had the opportunity to ask questions and all my questions have been answered to my satisfaction and understand the description of the research that is being provided to me.
- I agree that my data is used for scientific purposes and I have no objection that my data is published in scientific publications in a way that does not reveal my identity.
- I understand that if I make illicit activities known, these will be reported to appropriate authorities.

- I understand that I may stop electronic recordings at any time, and that I may at any time, even subsequent to my participation have such recordings destroyed (except in situations such as above).
- I understand that, subject to the constraints above, no recordings will be replayed in any public forum or made available to any audience other than the current researchers/research team.
- I freely and voluntarily agree to be part of this research study, though without prejudice to my legal and ethical rights.
- I understand that I may refuse to answer any question and that I may withdraw at any time without penalty.
- I understand that my participation is fully anonymous and that no personal details about me will be recorded.
- I understand that if I or anyone in my family has a history of epilepsy then I am proceeding at my own risk.
- I have received a copy of this agreement.

**PARTICIPANT'S NAME:**

**PARTICIPANT'S SIGNATURE:**

**Date:**

**Statement of investigator's responsibility:** I have explained the nature and purpose of this research study, the procedures to be undertaken and any risks that may be involved. I have offered to answer any questions and fully answered such questions. I believe that the participant understands my explanation and has freely given informed consent.

**RESEARCHER CONTACT DETAILS:** [mulherp@tcd.ie](mailto:mulherp@tcd.ie)

**INVESTIGATOR'S SIGNATURE:**

**Date:**

**PARTICIPATION:**

You will be invited to participate in a face to face or video conferencing interview.

Participation is voluntary and you may withdraw at any time and for any reason without penalty.

You will be asked if this interview can be recorded. Only audio will be recorded for face to face and video conferencing interviews. If you do not want it recorded, no audio recording

will take place. Instead I will transcribe your answers.

You have the option to decline to answer any questions that are asked. You have the right to withdraw from the research at any stage without penalty.

Regardless of recorded or transcribed interviews, the information you supply will be treated in accordance with the Data Protection Act. All answers will only be available to the researcher will only be used the purpose of this research.

The researcher is looking for the participants' personal opinions and not the opinions of the participants' organisation.

## Appendix 4 - Interview Questions

### Participation

Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.

Interview Questions

### Background

Q1. What does your organisation do?

Q2. What industrial sector do you work in and how many people does your company employ in software development?

Q3. Are you familiar with agile development methods for software development?

What roles and responsibilities have you held in the context of using agile methods?

How many years' experience do you have with agile?

Q4. Have you participated in software development project(s) using agile methods where the software development teams worked in two or more geographic locations? Describe the project(s)

How many projects have you delivered using Agile and Distributed Teams?

What are/were the geographic locations of the teams?

What was/is the time difference?

What was its impact on the way you or the team worked?

### Process or Control

Q5. What Agile methodologies did you use for distributed development projects? i.e. Kanban, Scrum – What was the reason for choosing these methods?

Are/were the project(s) delivered successfully?

Was their significant weight placed on an on timely completion of the project?

Was their significant weight placed upon project quality?

Was their significant weight placed upon project completion to meet client requirements?  
How?

Are/were the projects a success/failure? Yes, No

What were the reasons for this success/failure?

Describe the control mechanisms in place for the project with remote teams?

Procedural Controls (Status reviews, tracking project milestones)

Social controls (Information exchange, intercultural workshops)

Hybrid controls (site visits, common understanding workshops)

How are staff recruited for the distributed development teams?

How is trust built up in the absence of face to face communication?

Q6. Typically how many people worked in the development teams including local and remote staff?

What roles were remote and why?

What roles were locally based and why?

How did you decide and what factors influenced the splitting of the roles between local and remote staff?

Based on your experience what would you have changed in terms of splitting the roles between local and distributed teams?

Did the distribution of the team roles and team size impact the success or failure of the project? Why?

How long are your sprints?

How are agile practices standardised across development teams?

Q7. What is the typical duration of the projects you deliver using distributed teams?

Does the duration of the project impact the decision to use distributed teams? Why?

Would you say that projects of shorter duration were more successful using Agile and distributed teams? Why?

Q8. Do you modify agile methods for distributed software development teams by using a combination of Agile and Waterfall methods or other methods? If yes, please explain the approach.

Does the project integrate testing into the development sprints/time-boxes?

Does the project have a test period that begins after the completion of all development?

What documentation is provided to developers?

Is there a product backlog?

Was it a success?

Would you use this modified approach again?

Q9. With Agile methods how and when do you collect your requirements for the distributed development team?

How and why is this different to collecting requirements for local software development teams?

Does the requirements gathering approach work? Why?

How do you manage technical requirements scope creep and change control?

### **Contracts**

Q10. Does your organisation use fixed price contracts with Agile and distributed development teams?

For example:

Fixed price and fixed scope? Why?

Or variable price and fixed scope? Why?

Other?

Does Agile work for fixed price & fixed scope contracts? Why not?

#### Communication & Team Cohesion

Q11. Do you conduct daily stand ups with the remote team?

How was unplanned communication with the remote team conducted?

Would face to face communication have improved the communications between local and remote teams?

Does the daily stand work well? How would you improve the communication process?

How do you resolve communication problems or disagreements?

How is domain or product knowledge shared with the remote team?

Q12. How are your key stakeholders and their feedback represented at the daily stand-ups using distributed development teams?

Do your key stakeholders work directly with software development teams directly or through a product manager?

How do you manage stakeholder feedback with distributed teams?

Are stakeholders involved or represented in the daily stands? If no then, why not?

Are there ways to improve stakeholder representation with remote teams?

Q13. What communication channels or technologies are used to facilitate knowledge sharing with distributed development teams? (Agile process, development standards)

What worked best for you? Why?

Q14. From your experience what software development methodology is best suited to manage distributed software teams? Why?

Q15. What are the biggest challenges you have encountered using distributed software development teams?

Teamwork?

Collaboration?

Communication?