

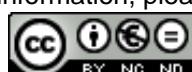


Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	A Study of Clan Control in Agile Software Development Teams
Author(s)	McHugh, Orla
Publication Date	2011-09-30
Item record	http://hdl.handle.net/10379/2188

Downloaded 2024-04-20T04:14:09Z

Some rights reserved. For more information, please see the item record link above.





**A Study of Clan Control
in Agile Software Development Teams**
by
Orla McHugh

B.Sc. (Applied Science), H.Dip. (Systems Analysis),
M.Sc. (Business Information Systems)

A research dissertation submitted in fulfillment of the requirements for the
degree of
Doctor of Philosophy
of the
National University of Ireland

Business Information Systems
J.E. Cairnes School of Business & Economics
National University of Ireland, Galway

Head of School: Dr. Emer Mulligan

Supervisors: Dr. Kieran Conboy
Dr. Michael Lang

Submission Date: September 2011

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Student ID Number: 03106942

Name of Candidate: Orla McHugh

Signature of Candidate:

Date:

Abstract

Agile methodologies have become an accepted choice of methodology in the information systems development community in recent years, but few studies exist that examine control in teams that use such methodologies. This study draws on control theory to understand how agile practices facilitate the resultant characteristics of clan control in agile software development teams. The nature of agile software development encourages flexible and adaptable software development teams, who are provided with substantial control over how they work. However, research that examines this self-control is scant, with even less addressing clan control in such teams. As a result, agile software development teams have little understanding of whether and how clan control exists in their team, or the impact agile practices may have on various characteristics of clan control.

This study first derives a framework from the literature that presents conditions required for clans to develop, the pre-conditions that should be in place in order to implement clan control, and resultant characteristics that emerge as a consequence of implementing clan control. It then proposes that three agile practices (sprint planning meeting, daily scrum meeting, and sprint review and retrospective meeting) help agile software development teams to achieve the inter-linked, resultant characteristics of clan control detailed in the framework.

A qualitative approach was used for this study. Three exploratory case studies were conducted to examine the application of the conceptual framework to agile software development teams. The teams selected were similar in size, from different industry sectors, and varied in their distribution. Primary data was gathered through 25 structured interviews and 14 observations of the agile practices across three teams in two different countries. Data was analysed for each agile practice using the seven resultant characteristics detailed in the conceptual framework.

Results show that the three agile practices facilitated each of the resultant characteristics of clan control, either wholly or partially, which indicates that clan control is in operation in these teams. However, it cannot be assumed that these three practices are the only contributors to the characteristics of clan control. These findings will be of benefit to practitioners who lack knowledge in the area of clan

control, or who are unaware of the impact agile practices may have on clan control within the team. Consequently, they may consider clan control as a suitable form of control in agile software development teams to supplement existing formal controls. Further studies should investigate the contribution of additional factors to clan control, such as other agile practices, the length of time the methodology was in use, the length of time individuals were employed by the organisation, the culture of the organisation, and the working environment.

One of the main contributions of this study is a framework that will guide future research on clan control. This is also a contribution to the management and control literature as this framework is not limited to agile software development and is suitable for application in many other areas. The study also adds to the current bodies of knowledge on clan control and agile software development.

The study concludes with recommendations for practitioners on how three agile practices studied can improve clan control in agile software development teams. These recommendations are made based solely on the findings from the teams studied and how these teams have implemented the agile practices. Future research can build on this to develop a more comprehensive set of recommendations.

Table of Contents

Abstract.....	ii
Table of Contents.....	iv
Index of Tables	ix
Index of Figures	xii
Explanation of Terminology	xiii
Acknowledgements.....	xv
1 Introduction.....	1
1.0 Background and Motivation.....	1
1.1 Research Objective	5
1.2 Structure of Thesis	5
2 Theoretical Foundation and Conceptual Framework.....	8
2.0 Introduction.....	8
2.1 Definition of Control.....	8
2.2 Control Theory.....	9
2.3 Control in Organisations	12
2.3.1 Control Frameworks	14
2.4 Clans	19
2.4.1 Clan Control.....	20
2.5 Development of Conceptual Framework	22
2.5.1 Identification of Elements for the Conceptual Framework	22
2.5.2 Conditions Necessary for Clans to Develop	24
2.5.3 Pre-Conditions Necessary to Implement Clan Control	26
2.5.4 Resultant Characteristics of Clan Control.....	31
2.5.5 Conceptual Framework	43
2.6 Conclusion	44
3 Agile Software Development.....	46
3.0 Introduction.....	46

3.1	Evolution of Software Development Methodologies.....	46
3.2	Agile Methodologies.....	50
3.2.1	Extreme Programming (XP)	54
3.2.2	Scrum	59
3.3	Agile Practices Investigated.....	63
3.3.1	Selection of Agile Practices	64
3.3.2	Sprint Planning Meeting	65
3.3.3	Daily Scrum Meeting	66
3.3.4	Sprint Review and Retrospective Meeting.....	68
3.3.5	Tailoring and Implementation of Agile Practices	69
3.4	Controlling Agile Software Development Teams.....	69
3.4.1	Agile Software Development Teams	70
3.4.2	Control in Agile Software Development Teams.....	71
3.4.3	Clan Control in Agile Software Development Teams	73
3.5	Conceptual Framework for this Study	74
3.6	Conclusion	77
4	Research Methodology	80
4.0	Introduction.....	80
4.1	Research Objectives.....	80
4.2	Epistemological Stance	81
4.2.1	Positivism.....	81
4.2.2	Interpretivism.....	82
4.2.3	Epistemological Stance for this Study	83
4.3	Quantitative Vs Qualitative.....	84
4.4	Research Approach	86
4.4.1	Case Study Method	88
4.4.2	Unit of Analysis	93
4.4.3	Single Case versus Multiple Case Studies	93

4.4.4	Research Approach for this Study	95
4.4.5	Validity and Reliability.....	96
4.5	Data Collection Methods	99
4.5.1	Interviews.....	101
4.5.2	Observation	104
4.6	Field Study Design.....	106
4.6.1	Case Selection	107
4.6.2	Theoretical Sampling and Saturation.....	108
4.6.3	Criteria used for Selection of Cases.....	108
4.6.4	Case Profiles	110
4.6.5	Design of Data Collection Protocol	116
4.6.6	Pilot Study.....	117
4.7	Data Collection	119
4.7.1	Planning Data Collection	119
4.7.2	The Interview Process.....	123
4.7.3	Direct Observation	126
4.8	The Process of Analysing and Coding the Empirical Data.....	131
4.8.1	Data Analysis	131
4.8.2	Coding.....	131
4.8.3	Coding process used in this study	132
4.8.4	Write-up	135
4.9	Summary	135
5	Findings and Analysis.....	137
5.0	Introduction.....	137
5.1	Sprint Planning Meeting	138
5.1.1	Members exhibit a strong sense of identity with the clan	138
5.1.2	Members exhibit a strong sense of commitment to the clan	150
5.1.3	Members develop a strong sense of loyalty towards the clan	156

5.1.4	Members develop common interests and goals which evolve as tasks progress	161
5.1.5	Rituals and ceremonies identify and reinforce acceptable behaviours.....	162
5.1.6	Members are influenced by shared norms, values and beliefs	169
5.1.7	Members evaluate and reward or sanction each other	174
5.2	Daily Scrum Meeting.....	175
5.2.1	Members exhibit a strong sense of identity with the clan	176
5.2.2	Members exhibit a strong sense of commitment to the clan	183
5.2.3	Members develop a strong sense of loyalty to the clan.....	187
5.2.4	Members develop common interests and goals which evolve as tasks progress	192
5.2.5	Rituals and ceremonies identify and reinforce acceptable behaviours.....	194
5.2.6	Members are influenced by shared norms, values and beliefs	201
5.2.7	Members evaluate and reward or sanction each other	206
5.3	Sprint Review and Retrospective	210
5.3.1	Members exhibit a strong sense of identity with the clan	210
5.3.2	Members exhibit a strong sense of commitment to the clan	216
5.3.3	Members develop a strong sense of loyalty to the clan.....	218
5.3.4	Members develop common interests and goals which evolve as tasks progress	222
5.3.5	Rituals and ceremonies identify and reinforce acceptable behaviours.....	224
5.3.6	Members are influenced by shared norms, values and beliefs	229
5.3.7	Members evaluate and reward or sanction each other	233
5.4	Summary	237
6	Recommendations	239
6.0	Introduction.....	239
6.1	Sprint Planning Meeting Recommendations.....	239
6.2	Daily Scrum Meeting Recommendations	249
6.3	Sprint Review and Retrospective Meeting Recommendations	258

6.4	Summary	265
7	Discussion and Conclusion	266
7.0	Introduction.....	266
7.1	Review of Research Objective.....	266
7.2	Discussion	268
7.2.1	Members exhibit a strong sense of identity with the clan	268
7.2.2	Members exhibit a strong sense of commitment to the clan	271
7.2.3	Members develop a strong sense of loyalty to the clan.....	272
7.2.4	Members develop common interests and goals which evolve as tasks progress	275
7.2.5	Rituals and ceremonies identify and reinforce acceptable behaviours.....	276
7.2.6	Members are influenced by shared norms, values and beliefs	276
7.2.7	Members evaluate and reward or sanction each other based on conformance to the values and beliefs of the clan	278
7.3	Contributions to Research and Practice	280
7.3.1	Contributions to Research.....	280
7.3.2	Contribution to Practice	281
7.4	Limitations	282
7.4.1	Limitations of the Study.....	282
7.4.2	Limitations of the Research Approach.....	283
7.5	Recommendations for Future Research	285
7.6	Conclusion	287
	References.....	289
	Appendix A – Interview Protocol	300
	Appendix B – Examples of Data Coding.....	309

Index of Tables

Table 2.3.1-1 Summary of control frameworks	19
Table 2.5.1-1 Clan control characteristics (Kirsch, 1996; 1997; 2004; Kirsch et al., 2010)..	24
Table 2.5.2-1 Conditions necessary for clans to develop.....	26
Table 2.5.3-1 Pre-conditions necessary to implement clan control	31
Table 2.5.4-1 Factors for identity.....	34
Table 2.5.4-2 Factors for commitment.....	35
Table 2.5.4-3 Factors for loyalty.....	36
Table 2.5.4-4 Factors for the development of common goals and interests.....	37
Table 2.5.4-5 Factors for rituals and ceremonies.....	39
Table 2.5.4-6 Factors for influenced by shared norms, values and beliefs	40
Table 2.5.4-7 Factors for team members evaluating, rewarding or sanctioning	42
Table 2.5.4-8 Resultant characteristics of clan control	43
Table 3.1-1 Values of Agile Methodologies	49
Table 3.1-2 Twelve Principles of Agile Methodologies	50
Table 3.2-1 Differences between plan-driven and agile methodologies (Nerur et al., 2005; Schuh, 2004, p8)	52
Table 3.2.1-1 Key Values of eXtreme Programming (Beck, 2000).....	56
Table 3.2.1-2 Key Practices of eXtreme Programming (Beck, 2000)	57
Table 3.2.1-3 Revised Practices of eXtreme Programming (Beck and Andres, 2005).....	58
Table 3.2.2-1 Key Values of Scrum (Schwaber and Beedle, 2002, p147).....	59
Table 3.2.2-2 Key Practices of Scrum (Elssamadisy, 2007; Schwaber and Beedle, 2002)...	60
Table 4.4-1 Relevant Situations for Different Research Methods (Yin, 2009, p8).....	86

Table 4.6.4-1 Profile of participating organisations and teams	115
Table 4.7-1 Data collection in the three cases studied	119
Table 5.1.1-1 SPM: Members exhibit a strong sense of identity with the clan	150
Table 5.1.2-1 SPM: Members exhibit a strong sense of commitment to the clan	155
Table 5.1.3-1 SPM: Members develop a strong sense of loyalty to the clan	161
Table 5.1.4-1 SPM: Members develop common interests and goals which evolve as tasks progress.....	162
Table 5.1.5-1 SPM: Rituals and ceremonies identify and reinforce acceptable behaviour..	169
Table 5.1.6-1 SPM: Members are influenced by shared norms, values and beliefs	174
Table 5.1.7-1 SPM: Members evaluate and reward or sanction each other.....	175
Table 5.2.1-1 DSM: Members exhibit a strong sense of identity with the clan.....	183
Table 5.2.2-1 DSM: Members exhibit a strong sense of commitment to the clan	186
Table 5.2.3-1 DSM: Members develop a strong sense of loyalty to the clan	191
Table 5.2.4-1 DSM: Members develop common interests and goals which evolve as tasks progress.....	194
Table 5.2.5-1 DSM: Rituals and ceremonies identify and reinforce acceptable behaviour .	201
Table 5.2.6-1 DSM: Members are influenced by shared norms, values and beliefs.....	206
Table 5.2.7-1 DSM: Members evaluate and reward or sanction each other	209
Table 5.3.1-1 SRRM: Members exhibit a strong sense of identity with the clan	216
Table 5.3.2-1 SRRM: Members exhibit a strong sense of commitment to the clan	217
Table 5.3.3-1 SRRM: Members develop a strong sense of loyalty to the clan.....	221
Table 5.3.4-1 SRRM: Members develop common interests and goals which evolve as tasks progress.....	223

Table 5.3.5-1 SRRM: Rituals and ceremonies identify and reinforce acceptable behaviours	229
Table 5.3.6-1 SRRM: Members are influenced by shared norms, values and beliefs	233
Table 5.3.7-1 SRRM: Members evaluate and reward or sanction each other.....	236
Table 5.4-1 Summary of Findings	238
Table 6.1-1 Recommendations for the Sprint Planning Meeting.....	247
Table 6.1-2 Recommendations for the Sprint Planning Meeting summarised by characteristic	248
Table 6.2-1 Recommendations for the Daily Scrum Meeting	256
Table 6.2-2 Recommendations for the Daily Scrum Meeting summarised by characteristic	257
Table 6.3-1 Recommendations for the Sprint Review and Retrospective Meeting	263
Table 6.3-2 Recommendations for the Sprint Review and Retrospective Meeting summarised by characteristic	264

Index of Figures

Figure 2.3.1-1 Conditions Determining the Measurement of Behaviour and of Output	16
Figure 2.5.5-1 Conceptual Framework: Clan Control	44
Figure 3.1-1 Waterfall Model Royce (1970)	47
Figure 3.2.1-1 Extreme Programming Process	55
Figure 3.2.2-1 The Scrum Process (Abrahamsson et al., 2002).....	61
Figure 3.5-1 Revised conceptual framework for this study	77
Figure 4.6-1 Case Study Method Adapted from Yin (2009, p57)	107
Figure 4.7.3-1 Scrum Board (Case C2).....	129
Figure 4.7.3-2 Burndown Chart (Case C2)	130
Figure 4.8.3-1 Data collected grouped by team in NVivo	132
Figure 4.8.3-2 Attributes defined for case study analysis	133
Figure 5.0-1 Conceptual framework used for this study.....	137
Figure 7.2-1 Conceptual framework used for this Study	268

Explanation of Terminology

This section contains a list of acronyms and terms that are used in this dissertation. An explanation of each term is provided for reference by a reader.

ASD	Agile Software Development
Burndown Chart	A Burndown chart is a publicly displayed chart showing remaining work in the sprint backlog that is updated every day.
Clan	A clan consists of a group of people who have a common goal, are dependent on one another, and where members of the group require social agreement as to what constitutes appropriate behaviour across a broad range of values and beliefs.
Clan Control	Clan control is the informal socialisation mechanisms that take place in an organisation and that facilitate shared values, beliefs, and understandings among organisational members.
Daily Scrum Meeting (DSM)	This is a short daily status team meeting lasting a maximum of 10-15 minutes typically conducted at the same time each day where members explain briefly what they accomplished since the previous meeting, what will be completed by the next meeting and indicate any impediments that may prevent them from completing their current tasks.
ISD	Information Systems Development
Iteration	An Iteration is a “time-boxed” period of work, with a closely defined and agreed output.
Planning Poker	This is a technique used by team members to collectively determine an estimate for each requirement (user story). It typically involves the use of a set of planning poker cards. An online version of planning poker is also available at www.planningpoker.com
Product Backlog	A Product Backlog is a high-level list of user stories that is maintained throughout the entire project. It contains a broad description of potential requirements for the software project.

Product Owner	The Product Owner leads the development effort by conveying his or her vision to the team, outlining work in the Product Backlog scrum backlog, and prioritizing it based on business value. The Product Owner typically represents the business (customer), is often a key user, and must be available to the team to answer questions and provide direction.
Scrum Board	A Scrum Board is a physical or electronic board that displays the work planned for the current sprint. It is updated daily during the daily scrum meeting with cards (tasks) moved across the board from the ‘To Do’ Column to the ‘In Progress’ column and finally to the ‘Done’ column.
Sprint	A sprint is a “time-boxed” period of work, with a closely defined and agreed output.
Sprint Backlog	A Sprint Backlog is the list of user stories the team work on during the next sprint (iteration) with each user story broken down into a number of hours work.
Sprint Planning Meeting (SPM)	This is a meeting that takes place at the start of each sprint where the team collectively defines and plan tasks that must be completed during the next sprint.
Sprint Review and Retrospective Meeting (SRRM)	This is a meeting that is held at the end of each sprint where the project team demonstrates the software completed to the customer and to management to obtain feedback. They then reflect on what went well in the sprint, what did not, and what could be improved for future sprints.
Timebox	A time box is a fixed amount of hours or days in which to accomplish something e.g. 2 weeks.
User Story	A user story is a short, clear description of the feature (requirement) that must be contained in the software system written using the language of the business user.
XP	eXtreme Programming

Acknowledgements

I wish to acknowledge the support and assistance of the following people in the completion of this dissertation:

I would like to thank the individuals in each of the three teams that facilitated my presence in their organisation for several days and who willingly gave their time to participate in this research. Without their cooperation and contribution this research would not have been possible.

I would like to acknowledge the support and assistance of my colleagues within the Business Information Systems group at NUI Galway over the past number of years.

Finally, I wish to express my deepest gratitude to my supervisors, Dr. Kieran Conboy and Dr. Michael Lang for their utmost diligence, constant and timely feedback, encouragement and guidance throughout the course of this research. Without their support and advice this dissertation would not have been possible.

This research was partly funded by grant aid received from the Millennium Research Fund at NUI Galway and by funding received from Enterprise Ireland as part of a wider European research project (Global Agile InnovAtion)

1 Introduction

1.0 Background and Motivation

Agile software development (ASD), which first emerged in the late 1990's, has changed the way in which software is developed and has resulted in a move away from heavy bureaucratic processes that require extensive planning and requirements definition, adherence to various procedures and the generation of reams of documentation in order to predict how and when a piece of software is delivered (Boehm, 2002). Agile methodologies, otherwise known as lightweight methodologies, promote the incremental development of quality software, which is delivered in short sprints (Highsmith and Cockburn, 2001). They embrace change, by allowing the requirements to change as the project progresses from one iteration to the next and are ideal for information systems development (ISD) projects where requirements are unpredictable (Highsmith and Cockburn, 2001; Williams and Cockburn, 2003). While there are many different agile methodologies, each created by different practitioners, they all follow the same fundamental principles and values defined by the Agile Manifesto (AgileAlliance, 2001). Some examples of agile methodologies are eXtreme programming (XP) (Beck and Andres, 2005), Scrum (Schwaber and Beedle, 2002), Crystal (Cockburn, 2004) and Feature Driven Development (Palmer and Felsing, 2002).

The flexibility of agile methodologies addresses many of the difficulties faced by organisations, such as changing business requirements, technologies and strategies, yet they provide several challenges from a management and control perspective (Nerur, Mahapatra and Mangalara, 2005). ASD teams typically work in an environment of high uncertainty where tasks are often unpredictable and must be flexible and able to respond and adapt quickly to changes (Beck and Andres, 2005; Cohn and Ford, 2003). They differ considerably from traditional software development methodologies in a number of ways. Instead of controlling an ISD project by implementing a command-control structure (between the project manager and the team) and adhering to various processes, agile methodologies focus on people, shared leadership and collaboration amongst team members (Jurison, 1999;

Nerur et al., 2005). Traditionally, the project manager has authority to issue commands and orders to their team (Druskat and Wheeler, 2003), is responsible for the delivery of the project, and is held accountable for the work completed by the team. Agile methodologies on the other hand rely on the team to self-organise and to share planning and decision-making (Jurison, 1999), be accountable to each other, and be collectively responsible for the delivery of the software project (Cockburn and Highsmith, 2001). ASD teams are empowered, with management having little control over their work, which represents a change in the balance of power that some project managers may find difficult to accept (Nerur et al., 2005). Consequently, new challenges may exist in such teams as to how the team manage and control a software project, which warrant investigation.

Agile methodologies have grown in popularity in recent years amongst practitioners and researchers with many organisations adopting an agile approach to software development (Dybå and Dingsøyr, 2008; VersionOne, 2009). ASD teams are purported to be flatter in structure than their non-agile counterparts, therefore requiring different forms of control similar to organisations that have moved away from the traditional hierarchical command and control structure to a flatter organisation structure that requires greater levels of cooperation and coordination horizontally across the organisation (van der Meer-Kooistra and Scapens, 2008). ASD teams are empowered to make decisions and have control over their own work (Schwaber and Beedle, 2002, p45). They set and comply with their own rules, define their own behaviours, rely on collaboration and communication, and place a strong emphasis on teamwork (Coram and Bohner, 2005; Nerur et al., 2005; Schwaber and Beedle, 2002). While a number of studies have addressed some of the management challenges organisations and project managers face when implementing an agile methodology (Augustine, Payne, Sencindiver and Woodcock, 2005; Boehm and Turner, 2005; Ceschi, Sillitti, Succi and De Panfilis, 2005; Coram and Bohner, 2005; Nerur et al., 2005) there is an absence of studies that use control theory to examine ASD projects (Maruping, Venkatesh and Agarwal, 2009).

There is ample research on agile methodologies, yet empirical research that applies control theory to ASD, or that examines agile practices from a control perspective is limited though there are a few notable exceptions. One is that by Harris et al. (2009)

who argue that formal controls in flexible projects are insufficient and present a new control called “emergent outcome control”, that describes a controlled-flexible approach to ASD that directs how outputs evolve during the project. However, they acknowledge that this type of control should be used as part of a wider portfolio of controls along with other formal and informal controls. A second study by Cram and Broham (2010) propose a new typology of control, which they use to distinguish between control dimensions within structured and flexible software development. Maruping, Venkatesh and Agarwal (2009) recognise that the study of ASD management is still in the early stages of growth. They use control theory to understand how agile practices contribute to an improvement in the quality of ASD projects and also call for further research that is more practice-focused (Maruping et al., 2009). Still, little research has focused on the impact of specific practices on teams and the associated factors that contribute to the success or failure of a project (Misra, Kumar and Kumar, 2009; Maruping et al., 2009). This is particularly relevant for practitioners who wish to adopt agile methodologies, but are unsure which practices or controls to implement that could benefit their teams.

Prior research that investigated control has focused on the ability of a manager to exercise control and observe the behaviours or outcomes of individuals, which are known as formal controls that are easy to monitor and measure (Ouchi, 1979; Ouchi, 1980). However, Kirsch (1996; 2004) argues that when control theory is applied to a complex, non-routine task such as the management of ISD projects, the type of control used may vary between formal and informal, and depends on the controller, the project, the individuals involved, and how they interact and relate to each other. In ISD projects, tasks and team members continuously change and evolve, such projects are often complex, and require participation from many diverse stakeholders, resulting in the utilisation of a number of different types of controls, both formal and informal, by management (Kirsch, 2004). ISD teams are composed of individuals with diverse personalities who contribute to the success or failure of an ISD project through their level of motivation, level of commitment, or how they deal with conflict or internal politics (Robey, Franz and Farrow, 1989; Markus and Benjamin, 1996). Consequently, the effective management and control of team members in such projects is imperative to ensure their continued positive contribution to a team

(Kirsch, 2004). Informal control such as clan control focuses on the socialisation of individuals who develop a common understanding of how they will work and interact with each other and consequently develop shared values and beliefs. Recent research found that the existence of clan control in ISD projects is related to project success and has been considered essential in such projects, yet developing clan control in an ISD project team is difficult as it takes time for clan control to develop naturally (Chua, Lim, Soh and Sia, 2007).

In ASD a group of individuals work together as a team (who can also be considered a clan) to achieve a common goal, where the team is in control, is flexible, and continuously adapts and reacts to changes in requirements, schedules, or risks (Harris et al., 2009). Within a clan each group member can effectively function as both the controller and the controllee (Choudhury and Sabherwal 2003). Such an environment can provide the conditions necessary for informal controls such as clan control to operate, as the use of formal controls, such as measuring and monitoring outcomes and behaviours, becomes more difficult. According to Ouchi (1979; 1980), where behaviours and outcomes are ambiguous and difficult to measure in organisations, clan control has been suggested as the most appropriate form of control. It is reasonable to assume that a similar principle can apply to ASD teams where behaviours and outcomes of teams are difficult to measure. This is somewhat supported by Cram and Brohman (2010) who illustrate that while a number of controls exist and are in use in ISD teams, there is initial support for the existence and usage of clan control in teams that use agile methodologies over other modes of control. It is also supported by Maruping et al. (2009) who indicate that ASD teams rely on informal controls such as clan control to assist with management and control as the team assume responsibility for the work they complete. This demonstrates that an empirical investigation of clan control in ASD teams is merited.

Empirical studies on Ouchi's notion of clan control are scant with few studies to date attempting to conceptualise clan control in the information systems literature (Chua et al., 2007; Kirsch, Ko and Haney, 2010). Instead, studies prefer to examine formal controls rather than informal controls. Several researchers have acknowledged the difficulty of conceptualising and operationalising clan control with calls for more studies on informal controls such as clan control (Eisenhardt, 1985; Rustagi, King

and Kirsch, 2008), particularly to study clan control at the level of the team (clan) because clan control focuses on the behaviour of a group of individuals working towards a common goal (Kirsch et al., 2010). This current study attempts to address these calls to some extent by investigating the development and existence of clan control in ASD teams. It has been acknowledged that it is difficult to develop clan control in ISD projects in a short space of time, but an ASD environment may be conducive to the quick development of clan control due to the nature of the methodology and the implementation of its practices. Agile practices encourage regular (daily) interaction and communication between team members, which helps to quickly develop relationships between members. As ASD teams focus on collaboration with people an extremely important component of the team, it is suggested that such teams are more effective as it is recognised that people who collaborate and work together are more productive than those who work solo (Cockburn and Highsmith, 2001). Prior research has identified a number of pre-conditions for clans to develop (Wilkins and Ouchi, 1983), for clan control to operate successfully, predominantly at organisational level (Eisenhardt, 1985; Ouchi, 1979; 1980) and has also identified various characteristics of clan control (Kirsch, 1996), which will be investigated in this study through the lens of agile practices.

1.1 Research Objective

The objectives of this study are as follows:

- (a) Develop a framework of the pre-conditions and resultant characteristics of clan control.
- (b) Use this framework to investigate how agile practices help ASD teams to achieve the resultant characteristics of clan control.
- (c) Develop a set of recommendations to identify how agile practices may be improved to achieve higher levels of clan control.

1.2 Structure of Thesis

This thesis is structured as follows:

Chapter 1 introduces the study, details the background and motivation for the study and outlines the research objectives.

Chapter 2 begins by defining control. It then presents a review of the literature on control theory and its application in organisations. A number of control frameworks are subsequently examined that focus on clan control. Various elements of clan control are identified and the chapter concludes with the presentation of a conceptual framework for clan control. This framework details the conditions necessary for a clan to develop, the pre-conditions that must be in place in order to implement clan control, and the resultant characteristics of clan control.

Chapter 3 first examines the evolution of software development methodologies. It then presents the literature in relation to agile methodologies, its principles and practices. This is followed by a more in-depth examination of two agile methodologies and their practices, which are the focus of this study. It further examines control in ASD teams with a specific focus on clan control in ASD teams. The chapter concludes with a presentation of the framework used for this study.

Chapter 4 commences by detailing the research objectives for this study. It then discusses the philosophical assumptions underpinning this research and reviews the debate in the literature between quantitative and qualitative studies. Details of the research methodology and data collection methods selected for this study are provided along with the limitations of these methods. The design of the field study is then presented, which incorporates the identification and selection of cases, the design of the research instrument, details of the pilot study conducted and the data collection process. The chapter concludes with a description of the process used to analyse the empirical data collected.

Chapter 5 presents the findings and analysis of the empirical data collected for each of the three agile practices studied using the resultant characteristics of clan control detailed in the conceptual framework as a lens for the analysis of the data gathered. The chapter concludes with a summary of the main findings for each of the resultant characteristics of clan control.

Chapter 6 details recommendations for practitioners on how each of the three agile practices studied may be improved in each of the three cases studied to achieve higher levels of clan control. These recommendations are made based on the findings from this study and good practices from the teams studied.

Chapter 7 concludes the research and discusses the findings in the context of the existing literature. Contributions for theory and for practice are presented. Finally, the limitations of the research are identified and recommendations for future research are presented.

2 Theoretical Foundation and Conceptual Framework

2.0 Introduction

This chapter presents the theoretical foundation for this study. Firstly, control is defined in section 2.1. This is followed by a review of the literature on control theory in section 2.2 and why control theory is relevant to this study. Section 2.3 examines control in the context of organisations and presents a number of existing control frameworks (section 2.3.1). Section 2.4 focuses on clans and clan control. Section 2.5 reviews the literature for existing clan control frameworks and identifies elements of clan control that can be incorporated into a framework which forms the basis for this study. This framework is refined in section 3.5 (p.74) for the purposes of this study. This section provides a detailed discussion on the conditions that must be in place for clan control to develop, the pre-conditions that must be in place in order to implement clan control and the resultant characteristics of clan control. Finally, the chapter concludes with a summary of the literature (section 2.6).

2.1 Definition of Control

“Control” is an ambiguous term with several researchers including Giglioni and Bedeian (1974) and Flamholtz et al. (1985) stating that control can have different meanings in different contexts. More recently, Collier (2005) presents several different synonyms for control; for example, “to direct”, “to govern”, “to exercise power over”, or “to command”. This section will examine the meaning of control in the context of organisations. Historically, organisations have adopted one of two approaches to control; namely, performance evaluation (referring to the cybernetic process of monitoring and rewarding performance), or focus on people policies such as training and socialisation (Ouchi, 1979; Blau and Scott, 1963). In hierarchical organisations rules and procedures are defined by senior management and communicated to lower-level employees who carry out the work (Ouchi, 1978). It is assumed that it is possible to measure the desired output to some degree of accuracy, which is essential in these hierarchical organisations, where specific measures and adherence to behaviours are used to verify performance (Ouchi, 1979). This results in two different ideas of control: one refers to domination or power where one person

(controller) attempts to influence another (controllee) while the other views control as regulation, where one individual governs or directs another according to specific rules and identifies areas of non-compliance (Emmanuel, Otley and Merchant, 1990, p7; Tannenbaum, 1968, p8). Therefore, the process of control can be defined as the process of monitoring behaviour, evaluating the outcomes that result from that behaviour, and providing feedback and reward (Ouchi, 1977; Ouchi and Maguire, 1975; Ouchi, 1978).

There are numerous definitions of organisational control, although most authors interpret organisational control as the influence of individual(s) over each other with the aim of achieving organisational goals. One such definition is that by Tannenbaum (1962), one of the foremost individuals associated with organizational theory and organisational control, who defines control as “*any process in which a person or group of persons or organization of persons determines, i.e., intentionally affects the behaviour of another person, group or organization*”. Similarly, Mills (1983) describes organisational control as “*a process designed to regulate the activities of organisation participants and their output*” and Flamholtz et al. (1985) define it as “*attempts by the organisation to increase the probability that individuals and groups behave in ways that lead to the attainment of organisational goals*”. All of these imply that individuals are influenced to make decisions and take actions that are consistent with the goals of the organisation. A number of other authors agree that control can be viewed broadly as *an attempt by individual(s) or organisations to influence people to take actions and make decisions, which are consistent with the goals and objectives of a group of individuals or organisations* (Das and Teng, 1998; Eisenhardt, 1985; Jaworski, 1988; Ouchi, 1979). This interpretation is used by several researchers that study control in ISD, which this research will also adopt.

2.2 Control Theory

Control theory is based in cybernetics, the science of control and communications, and is a general approach to the understanding of self-regulating systems. Cybernetics was originally developed by Norbert Weiner in 1948 to study communication and control processes in machines, living things, and social systems. Control theory has been in existence for years with researchers studying control in

many disciplines such as engineering (Dransfield, 1968), mathematics (Engelberg, 2005; Jacobs, 1974) and psychology (Carver and Scheier, 1999). Control has also been a central focus in the study of organisations (Mills, 1983) and is recognised by Eisenhardt (1985) as an important aspect of an organisation. It has also been applied in the information systems domain and in particular on ISD teams (Choudhury and Sabherwal 2003; Cram and Brohman, 2010; Harris et al., 2009; Henderson and Lee, 1992; Kirsch, 1997; Kirsch et al., 2010; Kirsch, Sambamurthy, Ko and Purvis, 2002). Maruping et al. (2009) and Harris (2009) further extend the application of control theory to ASD.

Control theory is an appropriate choice of theory in the study of organisations, teams, and specifically ISD teams for a number of reasons. Firstly, organisations have become more complex. Since the 1980's the managerial literature has been detailing the demise of the hierarchical organisation and putting forward the idea of a flat organisational structure (Barker, 1993). Otley (1994) echoes this and believes that the outcome of these studies are now dated as many modern organisations are smaller, flatter in their organisational structure, with greater interdependencies. This was predicted by Drucker (1988) who indicated that the modern organisation would be knowledge-based or information-based, where employees direct their own work and obtain feedback from each other and from customers. In the 1990's market demands began to change rapidly and these modern organisations needed to adapt and change quickly to meet the demands (Otley, Broadbent and Berry, 1995). This rapid change in market demands was contributed to by increased global competition, advances in technology, decisions to concentrate on 'core' business, the decline of manufacturing in the Western world, and a change in the expectations of society and its consumers (Otley, 1994; Otley et al., 1995). The increase in uncertainty and the decrease in manufacturing, where traditional management control techniques were implemented, began to pose difficulties for organisations from a control perspective (Otley, 1994). Simons (1995a) questioned how organisations that are expected to be innovative, creative and flexible can exercise adequate control over their employees. While formal, mechanistic control is appropriate in some organisations e.g. assembly line work (Simons, 1995a) teams in modern organisations are cross-functional, given autonomy, and encouraged to take control and responsibility for the work they do in

order to attain the goals of the organisation (Berry, Coad, Harris, Otley and Stringer, 2009; Otley, 1994). These organisations employ knowledge-based workers and must give lower management authority and responsibility for management and control, removing the command-and-control structure that existed previously (Drucker, 1988). This has resulted in the creation of work groups who have increased levels of self-control and group accountability, which means that performance and accountability may be evaluated differently or different control mechanisms may be utilised to measure performance (Otley, 1994). This can result in the emergence of a social organisation where the employee's beliefs, their ways of acting, of thinking, and in particular of interacting with one another become similar due to the social conditions they find themselves in even though they have not deliberately set out to form a social organisation (Blau and Scott, 1963, p1).

Secondly, the use of teams in organisations has increased dramatically as has the study of teams in organisations (Cohen and Bailey, 1997). Teams comprise of groups of individuals that work together, may have a diverse range of knowledge and skills, are dependent upon one another, have shared responsibility for outcomes, and have one or more tasks to perform in order to accomplish various goals (Cohen and Bailey, 1997; Guzzo and Dickson, 1996; Hackman, 1990; Mayer, Davis and Schoorman, 1995). Consequently, organisations face a number of challenges such as determining how team members will work together collaboratively to achieve the overall goal of the team or how to accurately measure the contribution of each team member (Towry, 2003). One way of overcoming these challenges is through control. While formal controls have a number of benefits they can be restrictive and can inhibit creativity with problems occurring if controls are too tight (Child and McGrath, 2001; Jaworski, 1988). Therefore, informal controls, such as clan control can be used as a tool to motivate team members to exhibit desired behaviours (Kirsch et al., 2010).

Thirdly, in the context of ISD control theory is an appropriate choice of theory to study the management and control of teams, which is reflected in the number of studies that use it to examine control with many frequently using Ouchi's (1977; 1979) control framework as a foundation. For example, studies have used control theory to examine control in outsourced software projects (Choudhury and

Sabherwal 2003; Rustagi et al., 2008), the control relationship between various stakeholders and team members (Henderson and Lee, 1992; Kirsch et al., 2002; Kirsch, 2004), the control process (Kirsch, 1996; Kirsch, 1997) and most recently control in open-source software development (Xu, Lin and Xu, 2011). The majority of these studies have focused on controls in general, or specifically on formal controls such as, behaviour and outcome controls with few examining the more subtle, informal controls.

2.3 Control in Organisations

Control has been a key concept in organisational theory and it is an important aspect of any organisation (Barker, 1993) as it is the process that ensures the conformance of members to the goals and objectives of the organisation (Tannenbaum, 1962). Control in organisations is largely based in management control theory although its roots are in systems theory and the theory of cybernetics (Collier, 2005). Organisations are social units, consisting of members, established with the explicit purpose of achieving specific goals (Blau and Scott, 1963, p1; March, 1965, p650). They have a high degree of structure and formality and controls must exist and be enforced to ensure that processes are followed and order is maintained to achieve the required goals (March, 1965, p650; Tannenbaum, 1962). It is generally the function of management to control employees and influence their behaviour to meet the goals of the organisation (Merchant, 1985, p4).

Much of the early research on management control focused on large hierarchical organisations with individuals assigned to lead a specific division or unit of the organisation who are then held accountable for the performance of the division or unit. Giglioni and Bedeian (1974) reported on one of the first studies conducted of control in a commercial organisation, which was published in 1941 by Holden, Fish and Smith. They discovered that control is one of the main responsibilities of senior management, who should identify the goals for the organisation, determine how to achieve these goals and evaluate how well the tasks were done once completed. March and Simon (1958) propose that the task itself determines which type of control should be implemented. For example, if a task is routine and it is easy to predict the outcome, then formal bureaucratic control should be exercised.

However, other types of controls can exist within organisations. One such example is an informal control called social control. It is in existence since the 1920's where social control is viewed as "*a general concept for describing all manner of activities involving the coordination, integration, regulation or adjustment of individuals or groups to some ideal standard of conduct*" (Chriss, 2007, p16). This description is based on a review of the work of two of the foremost sociologists Emile Durkheim and Edward Ross who emphasise the idea of social control as regulation where individuals who do not comply with the rules (social norms) of society are punished by society. Flamholtz et al. (1985) argue that organisational culture is a form of social control with employees developing a similar set of values and norms, consistent with those of the organisation, through the process of socialisation. Fortado (1994) and Collier (2005) too recognises the importance of social control in organisations with Collier (2005) criticising the lack of attention to social controls in both Simons (1995b) management control framework and Ferreira and Otley's (2005) performance management and control framework. Other researchers also criticise these frameworks for their narrow view of control, which focus on the conventional aspects of control (monitoring internal processes and actions), and ignore the social and behaviour aspects of control (Berry, Broadbent and Otley, 2005; Macintosh, 1994; Herath, 2007).

The importance of informal controls is also evident in a study by Jaworski et al. (1993) who identified that managers should be aware of the informal controls that exist within their teams and of how much they themselves can use such controls to influence the team to encourage morale and group cohesiveness. They are also used to encourage employees to conform to expected behaviours or to face sanctions for non-conformance to behaviours (Fortado, 1994). Studies in the accounting domain are increasingly acknowledging the importance of informal controls in organisations to encourage information sharing, learning and interaction amongst individuals (van der Meer-Kooistra and Scapens, 2008). However, it is still agreed that formal controls, or a reduced number of formal controls are required (van der Meer-Kooistra and Scapens, 2008) with Collier (2005) viewing control as a combination of formal, system-based controls and informal, social-based controls.

2.3.1 Control Frameworks

In an attempt to identify an existing framework that was suitable for this study an examination of the literature was conducted for control frameworks. Several frameworks were identified that focused on control at organisational level, rather than at department or team-level. They also tended to focus on control at a broad level rather than on a single specific control.

For the most part the control frameworks studied divided control into two main categories: formal control and informal control. Formal control specifies rules and procedures that require particular patterns of behaviour to be followed in order to achieve desired goals; behaviours or outcomes are measured and evaluated; and rewards are allocated in accordance with the tasks achieved (Das and Teng, 1998; Eisenhardt, 1985). Informal controls differ from formal controls in that they are social or people-based and they focus on the role that individuals or groups play in the exercise of control (Eisenhardt, 1985; Jaworski, 1988; Ouchi, 1980). They are typically instigated and implemented by employees and are unwritten rules that influence how employees behave (Jaworski, 1988). Some of the frameworks frequently cited in the literature are presented here.

Hopwood (1974) presents one of the first frameworks that extends the concept of control from formal accounting-based controls to consider informal controls such as social control and self-control as he believes that organisational control cannot be implemented without considering the social norms of the organisation. This was one of the earliest extensions of control outside of the management accounting domain. His framework details three types of control that exist within organisations:

- *Administrative Control*, which refers to the formal procedures that exist within an organisation that employees must adhere to. This incorporates accounting-based controls.
- *Social Control* - encourages employees to adapt similar values, norms and beliefs that match the values, norms and beliefs of the organisation
- *Self-Control* – encourages employees to establish their own behaviours and actions

This was later followed by Ouchi's seminal work (1979; 1980) which uses a transaction cost perspective to examine organisational control and predict the conditions under which a manager may exert a particular type of control. Ouchi (1978) believes that formal controls are only part of the control process within organisations and that employees experience informal controls before they are introduced to formal controls. For example, employees are identified and selected based on their skills and attitudes, which meet the needs of the organisation; they are provided with training; and are assigned to a team who socialises them into the organisation over a period of time. Ouchi's (1979; 1980) framework defines three types of control in organisations that he considers are equally important. Even though they can overlap with each other, he recognises that in some situations one type of control may be preferred over another. These three controls are:

- *Markets* – uses pricing information of products or services as a mechanism to regulate control, as these can be measured precisely. Employees are rewarded based on their personal contribution.
- *Bureaucracies* – involves the implementation and use of rules and procedures and the monitoring of employees to ensure that they comply with the rules.
- *Clans* – informal social structure where groups of individuals monitor themselves according to their values and norms.

Each of these three controls requires different information to operate. For example: markets require prices; bureaucracies require rules and procedures; and clans require traditions, common norms and values. Ouchi (1977; 1979) argues that the optimal choice of control is determined by characteristics such as task programmability (i.e. managers knowledge of the transformation process) and the ability to measure either the output or behaviour of employees. If an organisation's desired result can be measured, then it is recommended that *outcome control* is used whereas if organisations know the precise behaviours and processes that will transform inputs into outputs, *behaviour control* (personal surveillance) is used. If both behaviours and tasks can be defined and are measurable then either behaviour or outcome control is appropriate. However, Ouchi's (1977; 1979) framework (Figure 2-1) also recognises that situations can arise where it is not possible to measure either

behaviours or outcomes. In these instances organisations must rely on less formal social controls such as *clan control* to control the behaviour of employees.

		Knowledge of the Transformation Process	
		<i>Perfect</i>	<i>Imperfect</i>
Ability to Measure Outputs	<i>High</i>	Behaviour or Outcome Control (<i>Markets/Bureaucracies</i>)	Outcome Control (<i>Bureaucracies</i>)
	<i>Low</i>	Behaviour Control (<i>Markets</i>)	“Clan” Control (<i>Ritual</i>)

Figure 2.3.1-1 Conditions Determining the Measurement of Behaviour and of Output

This conceptual framework has become synonymous with the studies of control in the organisation and accounting literature. Subsequent research has applied (Eisenhardt, 1985; Jaworski et al., 1993; Snell, 1992) and extended (Govindarajan and Fisher, 1990) the framework in several different domains. In management accounting the literature has evolved to include other types of controls such as informal, personal and social controls, which differ from the traditional focus on financial controls (Otley, 1994). These informal and social controls are included in concepts presented by Merchant (1985, p17) and Emmanuel et al. (1990, p112) who classify control into the following three categories, similar to the classifications in Ouchi’s framework:

- *Results Control* – focuses on results (what Ouchi refers to as outcome control). Standards and metrics are defined, performance is measured against the standard and rewards are made based on the achievement of desired results. This type of control is common at management levels in most organisations, particularly large organisations, although it also permeates through organisations to the lowest level (Merchant, 1985, p17). If designed correctly, these controls are effective and at the same time motivate employees to achieve desired results (Merchant, 1985, p19).
- *Action Controls* – focuses on the behaviours of employees to ensure that they behave in a manner that is desirable and acceptable. With these controls steps are

taken to ensure that employees act in the best interests of an organisation (Merchant, 1985, p29). Conformance to policies and procedures is monitored and rewards/sanctions are based on compliance/non-compliance with policies and procedures (Emmanuel et al., 1990, p112).

- *Personnel controls* – this category combines both self-control, where employees naturally have a desire to do a good job, and social control, where employees are under pressure from the colleagues to conform to accepted group norms and values. Where self-control exists management rarely implement controls as employees are sufficiently self-motivated to achieve the desired outputs (Merchant, 1985, p39). However, management may also encourage groups of employees to monitor each others behaviour and influence each other, through the use of social control (what Ouchi refers to as clan control) in order to achieve the desired results (Merchant, 1985, p40).

These authors are also of the opinion that while each form of control is important, a single control is rarely sufficient on its own, or is appropriate for a particular situation, with each control or multiple controls required for different situations (Emmanuel et al., 1990, p113; Merchant, 1985, p19). Similar to Ouchi (1977; 1979) they view personnel control in terms of input mechanisms such as the recruitment or selection of suitable employees; the provision of training for these employees to ensure they understand their role within the organisation; the establishment of cohesive teams with common goals; the encouragement of peer control amongst these teams; and the creation of a strong corporate culture with which employees can identify.

Macintosh (1994, p132) further presents a framework based on Max Weber's seminal work on the sociology of organisations which identifies five different generic types of control and details the circumstances under which each type of control is most appropriate:-

- *Bureaucratic control* - uses formal mechanisms such as observations, rules and procedures to implement control. They work well when goals are clear and unambiguous, but are not as useful where goals change or are not well understood.

- *Charismatic control* – where leaders provide instructions to subordinates, but do not monitor subordinates as they know the work will be completed as expected. Even though leaders do not have formal control over their subordinates, subordinates are extremely loyal to their leader and complete the tasks requested without question even though the request(s) may take a long time to complete. This type of control is rarely successful in business as organisations pursue clearly-defined goals, which must be achieved.
- *Market control* – organisations use the price of products/services to regulate the behaviour of employees. It is a useful control when profit is the clear goal of the organisation. However, some form of bureaucratic control is also required to monitor profit targets across business units or division.
- *Collegial Control* – this is a form of control that exists in many institutions (often public-sector organisations) where decisions are made collectively by the group. It relates to control by a group of colleagues who are experts in a particular field, are of similar social status or education level. The group of colleagues have the authority to create rules and procedures to govern the actions of administrators, provide instructions to administrators, and monitor their work to see that tasks are completed, which can cause conflict between administrators and the group.
- *Control by Tradition* – Over a long period of time ideas are passed from one individual to the next, which become traditions that should not be challenged or questioned. One form of control by tradition is known as “clan control”, which can be a powerful form of control with members holding similar beliefs and values and having a strong sense of solidarity to the clan. This is a useful type of control when goals are ambiguous and not well understood.

Like Hopwood (1974), Ouchi (1979; 1980), Merchant (1985, p17) and Emmanuel et al. (1990, p112) Macintosh too recognises the importance of markets and informal controls such as social controls where there is a need to consider people and the environment in which they work in addition to formal administrative, bureaucratic controls. Each of these control frameworks are summarised chronologically in Table 2.3.1-1 and are categorised into the two broad categories of control, formal and informal. Note that the terms differ amongst the authors; for example, ‘outcome

'control' is also labelled 'results control', 'bureaucratic control' and 'bureaucracy'. Also, what Hopwood (1974) labels as 'adminstrative control' is used to refer to what others define as two distinct forms of control. The purpose of this table was to demonstrate that there is no single definitive control framework that presents all controls, rather a number of controls exist that can be interpreted differently. The next section will examine clan control, which is the focus of this study.

Authors		Hopwood (1974)	Ouchi (1977; 1979)	(Ouchi, 1980; Ouchi and Price, 1993)	Emmanuel et al.(1990) Merchant (1985)	Macintosh (1994)
Literature		Accounting Literature	Organisational Literature	Organisational Literature	Accounting Literature	Accounting Literature
Formal Control	Outcome	Administrative	Outcome	Bureaucracy	Results	Bureaucratic
	Behaviour		Behavioural	Markets	Action	Market
	Charismatic					Charismatic
Informal Control	Self	Self	-	-	Personnel	-
	Clan	Social	Clan	Clan		Control by Tradition
	Collegial	-	-	-	-	Collegial

Table 2.3.1-1 Summary of control frameworks

2.4 Clans

A clan consists of a group of people who have a common goal, are dependent on one another, and where members of the group require social agreement as to what constitutes appropriate behaviour across a broad range of values and beliefs (Das and Teng, 1998; Ouchi, 1979; 1980; Wilkins and Ouchi, 1983). Members of a clan have similar beliefs and interests with rituals and ceremonies playing an important part in the clan (Macintosh, 1994, p139). Members determine what is in the best interest of the clan and work towards achieving that (Wilkins and Ouchi, 1983). There is a high level of commitment on the part of each member of the clan to follow socially

prescribed behaviours, but these socially prescribed behaviours are often subtle and not always visible to those who are external to the clan (Ouchi, 1979).

A clan has a leader, who is not necessarily considered as the superior individual within the clan. Instead, the leader is seen as the chief, who is granted the ultimate authority for the clan (Macintosh, 1994, p138). Membership in clans can be highly exclusive and limited to certain individuals (Macintosh, 1994, p138). It is a lengthy process for new members to become socialised into a clan as it takes a long time to understand the values of the clan and become familiar with the social behaviours that are considered acceptable or unacceptable to the clan (Macintosh, 1994, p138). Clans, therefore, rely on the careful selection of individuals, who are socialised into the group, are very dependent on one another and work together to achieve a common goal (Ouchi, 1980). The selection of appropriate individuals in a clan is important as an organisation needs to be confident that they have the personnel who are capable of completing the work and that they can trust individuals to behave in a way that is consistent with the goals of the organisation (Ouchi, 1979). Members of the clan do not formally measure each others performance; instead, they subtly evaluate each other as they work together (Ouchi, 1980) and they believe that in the long-term rewards will be fairly and equitably distributed amongst the clan (Alvesson and Lindkvist, 1993; Wilkins and Ouchi, 1983).

2.4.1 Clan Control

The literature is consistent in its view of clan control. It has been described as a “people” or social process that is used by organisations when it is not possible to measure outputs and when behaviours are also unknown (Ouchi, 1977). The literature presents a number of definitions of clan control which include “*the use of social characteristics, such as shared values, commitment, traditions, and beliefs to control behaviour*” (Daft, 2009) or “*the informal socialisation mechanisms that take place in an organisation and that facilitate shared values, beliefs, and understandings among organisational members*” (Turner and Makhija, 2006). Clan controls are implemented by promoting a set of common values and beliefs within the organisation (Gopal and Gosain, 2010). Clan control exists when groups of individuals (clans) exhibit behaviour that is motivated by shared values and norms;

where individuals attempt to be “regular” members of the group by behaving in a manner that is consistent with agreed-upon behaviours; and the group (clan) reward or punish their members as they deemed appropriate (Ouchi, 1979). Clan control results in the exertion of social control on a group of individuals, otherwise known as a clan, to identify with the groups and its values (Berry et al., 2005, p25). It has also been called personnel control Merchant (1985, p4), input control (Cardinal, 2001; Snell, 1992), culture control (Merchant, 1985; Wilkins and Ouchi, 1983), or corporate culture (Alvesson and Lindkvist, 1993), although Wilkins and Ouchi (1983) take the classical view of culture and call it a clan. Turner and Makhija (2006) take a broader view of clan control and suggest that by using clan controls members develop a broader understanding of their own work and how it relates to the work of other members; they have greater flexibility in how they do their work; and they have the ability to change the target or the goal if it is deemed appropriate to do so.

Clan control is not always the control of choice for organisations, but under certain conditions it is used by some organisations with Wilkins and Ouchi (1983) claiming that in some organisations clan control is the dominant form of control. For example, Ouchi (1980) and Wilkins and Ouchi (1983) report on work conducted by James Abegglen on Japanese firms who found that they tended to hire inexperienced people, socialise them into the organisation and reward them based on criteria such as their length of service and their number of dependents. Japanese organisations rely heavily on clan control to control their employees. They do not need to exert outcome or behaviour control on their employees as each employee, who has become part of the clan, innately has a desire to do what is best for the organisation and believes that they will be fairly rewarded for their efforts. As all organisations hire personnel of varying levels of experience at many different levels the circumstances described are not representative of most organisations. A second example where clan control is the dominant form of control in an organisation is the United States Foreign Service Officer Corps. In this organisation it is difficult to measure how people can improve foreign relations, but a suitable selection and training process can help to choose the right people to work on the task (Ouchi, 1977). This view is supported by Daft (2009, p354) who believes that clan control exists in organisations with strong cultures of trust and shared values among

employees. This type of control is desirable when organisations or work is complex, uncertain, or unpredictable, as it may not be possible to specify rules and regulations, which makes it difficult to adhere to formal, bureaucratic controls (Ouchi, 1979). Chriss (2007, p56) argues that informal control, such as clan control, is only dominant in small groups at local level, but clan control in this instance may not be consistent with the goals of the organisation (Gopal and Gosain, 2010). Yet, where clan control does exist and is effective, there are a number of social mechanisms at play that result in a strong sense of community between the individuals in the clan (Ouchi, 1980) and it is important that managers do not try to impose other bureaucratic forms of control (outcome and behaviour), which can eradicate or disrupt this form of control (Ouchi, 1979).

2.5 Development of Conceptual Framework

This section develops the conceptual framework used as the foundation for this study. It is informed by literature in the following disciplines: organisational control, management accounting, and ISD. It also draws on a small number of studies on control in marketing and human resource management as these are cited by a number of researchers who investigate clan control. It first examines the literature to identify existing clan control frameworks and the various elements of clan control. It then progresses to develop a consolidated framework of clan control that incorporates the conditions required to develop clans (section 2.5.2, p.24), pre-conditions necessary to implement clan control (section 2.5.3, p.26) and the resultant characteristics of clan control that emerge as a consequence of these pre-conditions (section 2.5.4, p.31). This framework is used to guide the research objectives and the development of the research instrument.

2.5.1 Identification of Elements for the Conceptual Framework

No single definitive framework was identified in the literature that comprehensively describes the characteristics or concepts of clan control with Kirsch et al. (2002) acknowledging that the measurement of clan control is difficult. There are few attempts to conceptualise clan control with most attempts at conceptualising control related to formal controls (Eisenhardt, 1985; Henderson and Lee, 1992; Rustagi et al., 2008). However, there are concepts of clans and clan control that are generally

agreed in the organisational control literature with various researchers basing their conceptualisations on Ouchi's (1977; 1979) framework and many of these concepts applied in different disciplines.

As the presence of clan control is dependent on the existence of a clan the initial step was to identify conditions under which clans develop within organisations. An examination of the organisational and management control literature identified three conditions necessary to encourage the development of clans within organisations, which would lead to social agreement throughout the organisation. These were a long history and stable membership, the absence of institutional alternatives, and interaction amongst team members (Macintosh, 1994; Ouchi, 1979; Ouchi and Price, 1993; Wilkins and Ouchi, 1983). They are discussed in more detail in section 2.5.2 on page 24.

Further examination of the literature indicated that several conditions must be in place in order to implement clan control and several characteristics that result from the implementation of clan control were also identified by researchers. Kirsch, who is one of the most notable authors on the study of control in ISD presented a number of these conditions across a series of papers, which are summarised in the following key papers (Kirsch, 1996; Kirsch, 1997; Kirsch, 2004; Kirsch et al., 2010). These papers use Ouchi's (1977; 1979) framework as a basis for the study of control, but they also examine the literature on clans and clan control e.g. (Alvesson and Lindkvist, 1993; Barker, 1993; Birnberg and Snodgrass, 1988; Eisenhardt, 1985; Jaworski, 1988; Orlikowski, 1991) to establish these conditions and characteristics. The pre-conditions and resultant characteristics identified in these papers form the basis for the development of the conceptual framework and are detailed in Table 2.5.1-1.

A further review of the literature on clans and clan control identified other pre-conditions to implement clan control and resultant characteristics of clan control. Traditions, rituals and ceremonies were detailed by a number of researchers as a condition that must exist to implement clan control (Macintosh, 1994; Ouchi, 1979; Ouchi, 1980; Robey and Markus, 1984). In addition, managers should create an appropriate environment for clan control to develop (Alvesson and Lindkvist, 1993;

Lebas and Weigenstein, 1986; Merchant, 1985; Turner and Makhija, 2006), and finally, a high level of interaction and communication is required in order to implement clan control (Barker, 1993; Flamholtz et al., 1985; Gopal and Gosain, 2010; Turner and Makhija, 2006). An additional resultant characteristic of clan control was also identified, namely, members that develop a strong sense of loyalty to the clan as a consequence of the implementation of clan control (Alvesson and Lindkvist, 1993; Macintosh, 1994, p138).

Pre-Conditions to Implement Clan Control	Resultant Characteristics of Clan Control
Careful selection of members and socialisation help to propagate common values and beliefs	Members exhibit strong sense of identity with the clan
Cultivates culture of common values, philosophy and beliefs to problems-solving within the clan	Members exhibit a strong sense of commitment to the clan
Lack of prescribed behaviours and outcomes for tasks; instead behaviours and outcomes evolve as tasks progress	Members develop common goals as tasks progress
	Rituals and ceremonies identify and reinforce acceptable behaviours among members of the clan
	Members of the clan are influenced by shared norms, values and beliefs
	Individuals monitor each other and are rewarded or sanctioned for acting in accordance with the clan's values and attitudes

Table 2.5.1-1 Clan control characteristics (Kirsch, 1996; 1997; 2004; Kirsch et al., 2010)

The next three sections examine in greater detail the conditions under which clans can develop, the conditions that must be in place in order to implement clan control, and the characteristics that result from the implementation of clan control, which culminate in the presentation of a conceptual framework for clan control (Figure 2.5.5-1, p.44).

2.5.2 Conditions Necessary for Clans to Develop

Wilkins and Ouchi (1983) recognise that it is more likely that professional groups or business units within an organisation will develop clan (cultural) control rather than organisations due to the length of time required to develop stable memberships and socialise individuals into the organisation. This is supported by Eisenhardt (1985)

who agrees, in the context of a retail organisation, that clan control can take a long time to implement, but she suggests that it can be an appropriate form of control where the selection, training and socialisation of new employees can prove less costly than the implementation of formal measurement systems. Snell (1992) also argues that the careful selection, training and socialisation of individuals into an organisation can help to avert problems with performance, although it is not possible to guarantee increased performance. Consequently, Wilkins and Ouchi (1983) and Ouchi (1979) suggest a number of conditions that encourage the development of clans in large organisations, or what they also term social agreement, although the existence of the conditions does not guarantee the successful performance of an organisation (Table 2.5.2-1).

- (a) ***Long history and stable membership*** – Membership of a clan is often exclusive and may be limited to certain individuals (Macintosh, 1994, p138). Clans require stability of their membership, as they function based on the development of similar attitudes, values and beliefs, all of which take time to evolve (Ouchi, 1979). The longer an organisational unit is in existence, ideally with the same employees, the more likely members of the unit are to develop social understandings and knowledge, which can be passed on to a newer generation of employees, which exists in many Japanese organisations (Ouchi and Price, 1993; Wilkins and Ouchi, 1983). As social knowledge is passed on from one generation to the next it can become what many term culture (Wilkins and Ouchi, 1983).
- (b) ***Absence of institutional alternatives*** – employees who are not exposed to the cultures or traditions of other organisations will easily become socialised into the cultures and social conditions of an organisation (Wilkins and Ouchi, 1983). This can be done by organisations if they only recruit at the lowest level of an organisation and promote internally and also ensure that those they recruit possess values that fit with the values of the organisation (Wilkins and Ouchi, 1983).
- (c) ***Interaction amongst members*** – where employees in a well-established group/business unit that have worked together for a long period of time, have little interaction with other employees or business units they can develop their own culture and become ingrained in it (Wilkins and Ouchi, 1983).

These conditions indicate that it can take a long time for a clan to develop (Wilkins and Ouchi, 1983). Consequently, these conditions are more appropriate in the development of clans (culture) in large, well-established, bureaucratic organisations that mass produce products and less appropriate in modern high-tech organisations that work in uncertain environments (Alvesson and Lindkvist, 1993). Modern organisations are less likely to be concerned with developing clan control at organisation level; instead functional or professional groups within the organisation who are assigned to work with each other are more likely to develop clan control (Wilkins and Ouchi, 1983).

Conditions for clans to develop	References
Long history and stable membership	(Macintosh, 1994; Ouchi, 1979; Ouchi and Price, 1993; Wilkins and Ouchi, 1983)
Absence of institutional alternatives	(Wilkins and Ouchi, 1983)
Interaction amongst members	(Wilkins and Ouchi, 1983)

Table 2.5.2-1 Conditions necessary for clans to develop

2.5.3 Pre-Conditions Necessary to Implement Clan Control

As has discussed in section 2.4.1 clan control is a form of control that is used in circumstances where it is not possible to measure either behaviours or outcomes (Ouchi, 1979; Ouchi, 1980), although many organisations use clan control in conjunction with other forms of control. This type of control is subtle, illusive and intangible and it can take a long time for members to become familiar with the behaviours that are deemed appropriate (Macintosh, 1994, p139). If an organisation is conducive to the development of clan control a number of pre-conditions must be present in order to successfully implement clan control (Table 2.5.3-1), which were identified in section 2.5.1, p.22 and are discussed below. How these pre-conditions relate to ASD is examined in section 3.4.2.

- (a) Members of the clan must be *carefully selected and socialised* into the clan to ensure they are aware of the common goals and values and values of the clan (Ouchi, 1980; Ouchi and Price, 1993). Socialisation may take a long time to ensure that the behaviour of individuals conforms to accepted norms and it relies

on a low turnover of membership (Macintosh, 1994, p138; Ouchi and Price, 1993). Socialisation into a clan is achieved by the selection or recruitment of appropriate individuals, or by the training of employees in the beliefs and values of the organisation, which result in the modification of the beliefs of employees in order to become accepted; for example, the army (Ouchi, 1979; Merchant, 1985, p40). The selection of correct members is important as members that will easily be socialised into the clan are more desirable than those who are not with less resources and effort required to maintain adequate levels of control (March, 1965, p655; Merchant, 1985, p40; Ouchi, 1979). The more effective the selection process, the less socialisation is required to induct an individual to an organisation (March, 1965, p657). If members of the clan are socialised correctly into an organisation, then their goals will become the same as the goals of the organisation (Ouchi, 1980). For some professionals (doctors, solicitors, and consultants) the profession itself is an extremely powerful socialising agent with members expected to adhere to the values and norms of the profession, which are defined in professional codes of ethics (Abernethy and Stoelwinder, 1995).

- (b) A ***culture of shared values and norms, philosophy and beliefs*** (also called cultural control) must be developed within the clan, which influence the behaviour of members (Ouchi, 1980). The importance of culture in organisations and its role in control systems is recognised by a number of researchers (Birnberg and Snodgrass, 1988; Flamholtz et al., 1985). Flamholtz et al. (1985) view organisational culture as a form of social control whereas Birnberg and Snodgrass (1988) distinguish between culture at organisational level and culture at national or societal level. They contend that there is a distinction between the two – one is where individuals bring a set of norms and values to the workplace and the other is where individuals develop a set of norms and values within their work environment. They also argue that while culture groups generally have the same norms and values, differences may exist internally within the culture groups that are not consistent with the norms and values of the group, which is not uncommon at national and societal levels. Jaworski (1988) argues that the types of control differ in relation to the size of the unit they represent i.e. self-control (individual), social (clan) control (small group), culture control (large

social unit such as an organisation). Ouchi (1979) differs in his view by referring to culture as a large number of citizens who are socialised into a group and have the same values and norms whereas he describes a clan as the socialisation of individuals within an organisation. Jaworski's (1988) description of cultural control, albeit at organisational level, incorporates many of the characteristics of clan control such as the development of rituals and norms of social interaction through which individuals become internalized into the organisation and understand the goals of the organisation. Alvesson and Lindkvist (1993) are in agreement that clans and culture are similar concepts with culture used by organisations when it is difficult to measure and monitor tasks in the traditional formal way. More recently clan control has also been called social control (Jaworski, 1988; Merchant, 1985) or professional control (Abernethy and Stoelwinder, 1995) with Berry et al. (2009) referring to social (clan) control as a type of control that includes informal cultures and systems that influence individuals who are part of the social system. Consequently, for the purposes of this research culture control and clan or social control are considered to overlap sufficiently and are deemed comparable to combine under one type of informal control called clan control.

A culture consists of a variety of elements, which includes norms, values, beliefs and customs and the culture of an organisation i.e. its values, beliefs and customs can be considered a guide for employees as to what is considered acceptable behaviour (Berry et al., 2005, p37; Birnberg and Snodgrass, 1988; Herath, 2007). They are traditions or unwritten rules agreed between employees that regulate the behaviour of employees (Fortado, 1994; Jaworski, 1988; Ouchi, 1980) and may be described as simply as "*the way we do things around here*", which can be a powerful form of control that influences the behaviour of people (Merchant, 1985, p42). For example, a norm may develop that the team will automatically work overtime to meet customer demand (Barker, 1993). As individuals become internalised on the appropriate and accepted norms for an organisation or a profession they are socialised into the organisation or profession (Covaleski, Dirsmith, Heian and Samuel, 1998; Orlikowski and Baroudi, 1991). At organisational level it can be very difficult to develop strong cultural control,

although one possible solution is to move personnel internally, so that they become socialised within the organisation while at the same time they develop a knowledge and appreciation for the work carried out in different parts of the organisation (Merchant, 1985, p42). However, one study by Barker (1993) illustrated that it is possible for an organisation and its employees to develop norms which in this case became more powerful than the formal control system that was used formerly. In this case employees were placed into self-managing teams, which was a new way of working for these employees. They were given autonomy over their work and were encouraged to make their own decisions with support from their manager only when requested by the team. The teams initially did not know how to operate, but over time they developed and defined a set of values and norms. However, ultimately these norms and values resulted in negative consequences as team members began to feel constrained by the norms and new temporary employees felt under pressure to conform to the norms of the team in order to acquire a permanent position. The informal (social) control system developed by employees, which Barker (1993) calls concertive control, resulted in the creation of more controls and stronger controls than had the previous formal control system. Employees subjected each other to these controls and rewarded or punished each other for not adhering to the controls with punishment for not conforming to the norms as severe as an employee losing their job.

- (c) Prior studies on the pre-conditions required to implement control focused on the ability of a supervisor to measure outcomes and observe behaviours, which relate to formal controls (Eisenhardt, 1985; Ouchi, 1979). However, according to Ouchi's (1977; 1979) control framework where tasks are not routine, are unpredictable and their outcomes are difficult to measure clan control is considered an appropriate form of control. *A lack of prescribed behaviours and outcomes for tasks* is not uncommon in an environment where clan control exists as in this type of environment behaviours and outcomes evolve over time as the tasks progress (Ouchi, 1979).
- (d) ***Traditions, ceremonies and rituals*** exist which contain information that conveys the values and beliefs of the organisation (Lebas and Weigenstein, 1986; Ouchi,

1979). Traditions are implicit rules that govern behaviour (Ouchi, 1980). This information is not written down anywhere; therefore, it is difficult for individuals outside of the clan to gain access to such information (Ouchi, 1979). Traditions and rituals reinforce the code of conduct expected and help to provide meaning to actions taken and are normally beyond reproach and should not be questioned (Macintosh, 1994, p138; Robey and Markus, 1984). Yet, some rituals are questionable; for example, organisations may systematically gather information that is never used, or request reports that are never read (Feldman and March, 1981). At the same time some rituals are conducted because of the social norms that exist; for example, the report produced in the previous example may be requested by a manager who must be in a position to produce hard evidence if required, to support a decision made (Feldman and March, 1981).

- (e) To implement clan control managers or leaders must **create an environment that encourages clan control** (Lebas and Weigenstein, 1986; Turner and Makhija, 2006). This can include the allocation of appropriate physical resources such as office space, but may also refer to the provision of tools, knowledge and information (Merchant, 1985, p43). Employees must be provided with the necessary resources to complete their work as the working environment of employees can influence their attitudes towards their employer, which can have consequences for an organisation such as the long-term loyalty of employees to the organisation or the likelihood that they will remain an employee (Alvesson and Lindkvist, 1993).
- (f) A **high level of interaction and communication** is required amongst members for clan control to exist (Barker, 1993; Flamholtz et al., 1985) where communication refers to the “*exchange of meaningful information*” (Päivärinta, 2001). This leads to the sharing of resources and ideas between members, which helps member to achieve the goals of the clan (Gopal and Gosain, 2010). Turner and Makhija (2006) echo this in their study of organisational controls and how different types of controls impact on the transfer of tacit knowledge and information within an organisation, where tacit knowledge is knowledge that an individual is aware of, but is difficult to transfer or exactly explain what it is to another individual (Polanyi, 1962). They argue that teamwork and regular

interaction amongst team members, either through observation or face-to-face communication, is extremely important for knowledge sharing and knowledge transfer and that clan control plays a vital role in this. Members also become aware of the competencies of others and become comfortable to have their work scrutinised by other members, which should reduce rework and improve the quality of outputs, although in this research while quality did improved, efficiency decreased, implying that increased collaboration caused greater overheads (Gopal and Gosain, 2010). Members of clans use the tacit knowledge gained from their socialisation into the clan and draw on their own diverse range of experiences to work together to solve problems and achieve their goals (Turner and Makhija, 2006). Group meetings are one such way for sharing and acquiring information as they allow team members to interact, learn and share problems and solutions (Walz, Elam and Curtis, 1993).

Pre-Conditions Necessary to Implement Clan Control	References
Careful selection and socialisation of members into the clan	(Abernethy and Stoelwinder, 1995; Macintosh, 1994; March, 1965; Merchant, 1985; Ouchi, 1979; 1980; Ouchi and Price, 1993)
Culture of shared values and norms, philosophy and beliefs must be developed within the clan	(Barker, 1993; Berry et al., 2005; Birnberg and Snodgrass, 1988; Jaworski, 1988; Merchant, 1985; Orlikowski and Baroudi, 1991; Ouchi, 1979; Ouchi, 1980)
Lack of prescribed behaviours and outcomes for tasks; instead behaviours and outcomes evolve as tasks progress	(Eisenhardt, 1985; Ouchi, 1977; 1979)
Existence of traditions, ceremonies and rituals	(Lebas and Weigenstein, 1986; Macintosh, 1994; Ouchi, 1979; 1980; Robey and Markus, 1984).
Create an environment that encourages clan control	(Alvesson and Lindkvist, 1993; Lebas and Weigenstein, 1986; Merchant, 1985; Turner and Makhija, 2006)
High level of interaction and communication is required	(Barker, 1993; Flamholtz et al., 1985; Gopal and Gosain, 2010; Turner and Makhija, 2006)

Table 2.5.3-1 Pre-conditions necessary to implement clan control

2.5.4 Resultant Characteristics of Clan Control

As detailed in section 2.5.1 (p.22) the exercise of clan control results in seven characteristics summarised in Table 2.5.4-8 (pg43). This section reviews the

literature in relation to each of these seven characteristics and identifies a number of factors that further explain each of these characteristics. The existence of each characteristic is first explored in the organisational literature and is then examined in the ISD literature, and where applicable in the ASD literature. Further detail on agile methodologies and ASD is provided in Chapter 3 (p.46).

During the literature review a number of factors were identified that describe the first four characteristics in the conceptual framework. Researchers are not in consistent agreement that there is a clear distinction between the concepts of identity, loyalty, and commitment. Lee (1971) is of the view that these concepts are intertwined and that it is difficult to analyse them separately as each concept implies some aspect of another concept; for example, identity implies some degree of loyalty. Similarly, other researchers recognise that there is a relationship between a number of these characteristics such as commitment and loyalty, but they also acknowledge that there are variances between the two concepts (Becker and Billings, 1993; Bishop, Scott and Burroughs, 2000; Mak and Sockel, 2001). This makes it difficult to clearly distinguish between the concepts for the purposes of this study. In an attempt to address this complexity a definition is presented for each characteristic and the factors as identified in the literature are presented, including the overlaps between each of the characteristics.

The last three characteristics on the conceptual framework are self-explanatory and the factors presented are based on the description of the characteristic. The factors for each of the seven characteristics are presented in Table 2.5.4-1 to Table 2.5.4-7, which are later used in Chapter 5 as a lens for the analysis of the data gathered.

- (a) Individuals ***exhibit a strong sense of identity*** with the clan (Kirsch, 1996) where identification is “*a psychological state wherein an individual perceives himself or herself to be part of a larger whole such as a work group, a team, or an organisation*” (Rousseau, 1998). It has also been described by Lee (1971) as “*a degree of belongingness or loyalty*” where these concepts are intertwined. Developing a sense of identity with a group of individuals, who may be diverse, or dispersed, has also been termed collective team identification by Van Der Vegt and Bunderson (2005), which they define as “*the emotional significance*

that members of a group attach to their membership in that group”. Individuals that identify with each other are likely to have **similar goals, values, and norms** and will **work and cooperate** as a team (Eckel and Grossman, 2005; Lee, 1971). Where individuals identify strongly with a clan, their own **self-interests decrease in favour of the interests of the clan**, they behave in a manner that fulfills the needs of the clan, and are **committed to following the norms of the clan** (Kanter, 1968; Wenzel, 2004). They are also **committed to the team and its goals** rather than (or in addition to) their own goals (Van Der Vegt and Bunderson, 2005). Strong identification with the team (clan) can enhance collaborative behaviours within the team (clan) and has been positively linked to communication, mutual effort and support, and team cohesion (Carmeli, Gelbard and Goldreich, 2005). Where individuals do not identify with the clan, this can result in a reluctance to adopt the common norms, values and beliefs that are inherent in the clan. The factors that describe identity are summarised in Table 2.5.4-1.

Software development is a social process that requires interaction and communication with a variety of stakeholders, including the customer, to carry out desired changes (Päivärinta, 2001). Identification with the team is important in order for a team to successfully work and cooperate with these various groups of individuals to achieve their assigned goals (Marks and Lockyer, 2005). A key role in ASD is that of the customer. Their involvement in the team and cooperation with the rest of the team has been identified as a critical success factor for ASD teams (Chow and Cao, 2008; Misra et al., 2009; Nerur et al., 2005). Agile methodologies have increased the importance of team identity over individual identity or role identity (Whitworth and Biddle, 2007b). It is even more important where teams are distributed, especially where distributed team members never meet, as they need to develop and maintain a common identity with each other (Braithwaite and Joyce, 2005). In ASD teams a common identity is developed through the use of rituals such as agile practices, continuous communication and interaction between team members, and the indication and assurance of progress towards the common goal (Whitworth and Biddle, 2007b). Collaboration between team members and the customer is also enhanced

by the frequent (ideally, face-to-face) communication and interaction and by the short incremental and iterative development cycles (Asproni, 2004).

Factor	
F1	Members feel part of the team
F2	Members are loyal to the team
F3	Team members have similar goals, values, and norms and are committed to following them
F4	Team members work and cooperate as a team
F5	Team interests emphasised over self-interests

Table 2.5.4-1 Factors for identity

- (b) Secondly, members ***exhibit a strong sense of commitment*** to the clan (team) where commitment can be defined as “*the process through which individual interests become attached to the carrying out of socially organised patterns of behavior which are seen as fulfilling those interests, as expressing the nature and needs of the person*” (Kanter, 1968). The level of commitment of an individual to an organisation relates to their level of identification with the organisation, level of involvement in the organisation and **level of loyalty** to the organisation (Bishop et al., 2000; Buchanan, 1974). A high level of interaction with the organisation can indicate a willingness to expend effort in order to contribute to the interests of the organisation (Mowday, Porter and Steers, 1982, p27). Where individuals display a deep commitment to an organisation, the necessity to explicitly monitor that individual is reduced (Ouchi, 1979). Organisational commitment can be characterised by “*(a) a strong belief in and the acceptance of the organisation’s goals and value; (b) a willingness to exert considerable effort on behalf of the organisation; and (c) a strong desire to maintain membership in the organisation*” (Mowday et al., 1982, p27). Team commitment, or commitment to a clan, can be defined in a similar manner where teams (clans) **develop goals and values that members may accept, members exert considerable effort on behalf of the team** (clan), and members may **exhibit a desire to remain a member of the team** (clan) (Bishop et al., 2000) even though members may experience high levels of commitment to one or more of these factors and not another. For example, a team member may accept the goals

and values of the team and work hard on behalf of the team, but may have no desire to remain a member of the team.

Individuals are committed to a team when they **feel a strong sense of identification with the team** and are willing to expend effort to achieve their common goal, which can be fostered by having a clearly defined goal and involving all team members in all phases of the project (Larson and LaFasto, 1989, p74). In ASD all team members are expected to participate in frequent communication and interaction with other team members (Asproni, 2004). This includes the customer whose commitment to the team is considered critical for the success of a project (Chow and Cao, 2008; Misra et al., 2009; Nerur and Balijepally, 2007). Their role involves discussing product features, planning, making decisions and prioritising functionality in conjunction with the rest of the team in addition to providing regular feedback to the team (Boehm and Turner, 2005; Nerur and Balijepally, 2007). Hence, their lack of involvement and commitment can cause concern for the team (Hoda, Noble and Marshall, 2011). This literature is summarised in Table 2.5.4-2.

Factor	
F1	Team develop goals and values that members accept
F2	Members exert considerable effort on behalf of the team (clan)
F3	Member exhibit a desire to remain a member of the team (clan)
F4	Members identify with the team
F5	Members are loyal to the team

Table 2.5.4-2 Factors for commitment

(c) Thirdly, members feel a **strong sense of loyalty** when they belong to a clan (Alvesson and Lindkvist, 1993; Macintosh, 1994, p138). Buchanan (1974) describes employee loyalty to an organisation as “*a feeling of affection for and attachment to an organisation*”. The amount of loyalty an employee has to an organisation can relate to their **level of identification** with the organisation (Mak and Sockel, 2001). A similar view could be applied in the context of the loyalty

of a member to a clan. Members of a clan who work closely together, are dependent on one another, and **complete tasks for the good of the clan as opposed to the benefit of one individual, develop a sense of belonging and solidarity** with the clan and a **strong team spirit** can develop (Alvesson and Lindkvist, 1993). This can be achieved through the provision of good working conditions, interesting and varied work, and frequent social interaction (Alvesson and Lindkvist, 1993). ASD teams have reported a sense of belonging to the team due to the frequent interaction, communication, and discussions amongst team members, which differs with a plan-driven approach to software development where interactions only occur when needed (Whitworth and Biddle, 2007b). The factors that describe a sense of loyalty are summarised in Table 2.5.4-3.

Factor
F1 Members feel an emotional attachment to the team
F2 Members identify with the team
F3 Members develop a sense of belonging and solidarity with the team
F4 Team interests are emphasised over self-interests
F5 Members develop a strong team spirit

Table 2.5.4-3 Factors for loyalty

(d) Where clan control exists organisational goals are often unknown at the start of a project, but instead evolve over time (Ouchi 1979). Where they do exist clan controls may not be aligned with the goals of the organisation (Gopal and Gosain, 2010). Instead, members of the clan, who are dependent on one another **develop common interests and goals, which evolve as tasks progress** (Kirsch, 1996; Kirsch, 2004; Turner and Makhija, 2006) and as members become socialised into the clan and become familiar with the norms and values of the clan (Alvesson and Lindkvist, 1993; Harris et al., 2009). Managers can strategically encourage particular behaviours and norms in an attempt to align the goals of the clan with those of the organisation (Gopal and Gosain, 2010). Successful teams (clans) can spend a substantial amount of time and effort determining a common purpose and developing common goals for the team towards which they strive (Katzenbach and Smith, 2005). These team goals evolve during the life of the team and may be different from those of the organisation or to individual goals (Katzenbach and Smith, 2005) with members willing to contribute to the goals of

the clan even though they may not be rewarded in the short-term (Alvesson and Lindkvist, 1993).

When clan control exists in ISD teams they have **autonomy over their goals and can decide how to achieve their goals**, although management will try to ensure that their goals are aligned with those of the organisation (Maruping et al., 2009). At the beginning of a software project, ISD teams rarely have a complete understanding of the software system that is built; instead this evolves over time as the team learn more about the expected behaviour of the system (Walz et al., 1993). Users or customers rarely have the same interests or motivations as developers and may have conflicting interests when making decisions on the development of a software application, which can become even more complicated where there are multiple different users (Robey and Markus, 1984). For example, a developer may recommend the in-house development of a software application over the purchase of a software package because the developer relishes the challenge of delivering a new software application (Robey and Markus, 1984). Most software projects have a common primary goal of delivering high quality software, on time and within budget (Misra et al., 2009). While this is also true for software projects that use an agile methodology the requirements in an ASD project may change frequently (Asproni, 2004). ASD teams have greater autonomy over how they achieve their primary goal and are given the authority and autonomy and responsibility on how best to meet the goal (Sutherland, Viktorov, Blount and Puntikov, 2007; Williams, 2007). The use of iterative development helps ASD teams (including the customer) to clearly define and work towards short-term goals with the customer providing regular feedback on what they expect from the developers (Asproni, 2004). The factors that describe the development of common goals and interests are summarised in Table 2.5.4-4.

Factor
F1 Common interests and goals develop, which evolve as tasks progress
F2 Members have autonomy over goals and how to achieve them

Table 2.5.4-4 Factors for the development of common goals and interests

(e) **Rituals and ceremonies identify and reinforce acceptable behaviours** among members of the clan with individuals attempting to be “regular” members of the group by behaving in a manner that is consistent with agreed-upon behaviours (Kirsch, 1996; Ouchi, 1980). Rituals and ceremonies “*reinforce the implicit, but well understood codes of correct conduct*” and once these are understood members know how to behave without being told (Macintosh, 1994, p138). They require the recurring collective participation of members and they help members to develop loyalty to the group (Kanter, 1968). They are a source of information for members as they communicate the values and beliefs of the clan to members, which may be in the form of rules and regulations that are considered acceptable by the clan (Kirsch, 1997; Ouchi, 1979). The socialisation process, as well as the rituals and ceremonies help members of the clan to identify and reinforce acceptable behaviours and where all members are aware of their identity there is a shared understanding of the norms, values and beliefs of the clan (Hegtvedt, 2005; Kirsch, 1997).

Rituals are habitual tasks, which may be repeated every day, and are important in maintaining team spirit and identity. In ISD each development phase can be viewed as a ritual that signifies that each action taken has a purpose and is acceptable to the team (Robey and Markus, 1984). As many rituals exist in software development it is important that those who interact with the process are aware of these rituals and the functions they perform to ensure that they can effectively participate in the process (Robey and Markus, 1984). A customer may become a ‘regular’ member of the team by participating in the rituals of the team, and adopting the shared norms, values, and goals of the team, in order to achieve the overall collective goal of the team (Kirsch et al., 2010). In ASD teams practices such as sprint planning meetings, the daily scrum and the sprint review and retrospective meeting can be viewed as rituals and ceremonies; for example, by holding a meeting at the same time, in the same place (Schwaber and Beedle, 2002, p119). These ceremonies help to build a culture within the team and also to bond the team together (Schwaber and Beedle, 2002, p119). Teams can use such rituals to promote and re-inforce behaviour that they consider acceptable (Kirsch

et al., 2010). The factors discussed in the findings chapter for this resultant characteristic are detailed in Table 2.5.4-5.

Factor
F1 Existence of rituals and ceremonies
F2 Rituals and ceremonies identify acceptable behaviours
F3 Rituals and ceremonies reinforce acceptable behaviours

Table 2.5.4-5 Factors for rituals and ceremonies

- (f) Clan control establishes a set of norms, values, and informal rules by which members must behave (Lebas and Weigenstein, 1986). Members of the clan are **influenced by the shared norms, values and beliefs** that exist, which results in a group of individuals who have common values and beliefs and who are committed to the clan (Kirsch, 1997). They believe that their interests are best served when every member focuses on the needs of the clan rather than their individual needs (Macintosh, 1994, p139; Kanter, 1968). Common values and beliefs suggests that members of the clan share a general orientation that is cooperative in nature, they develop common interests for the clan, and an assurance that the personal goals of members are compatible with the goals of the organisation (Birnberg and Snodgrass, 1988; Ouchi, 1980; Wilkins and Ouchi, 1983). However, social norms affect the behaviour and beliefs of individuals where norms refer to the basic social agreement between members of a clan that allow it to function efficiently, without undue costs of monitoring adherence to the norms (Ouchi, 1980). Where norms are established and deviations from those norms occur the clan will attempt to subtly correct the deviation, but where members repeatedly deviate from the norms the clan may ostracise the offending individual (Jaworski, 1988). It is not sufficient to have shared norms, values and beliefs as this does not indicate the existence of clan control. It is when behaviour is influenced by these norms, values and beliefs; for example, they behave in a manner that is cooperative, collegial and in line with the expectations of the group, then clan control is in operation (Kirsch et al., 2002).

In ISD teams many different norms can develop, which influence the behaviour of team members. For example, a norm may develop within the team that team

members are expected to attend meetings, or that the customer expects the creation of high-quality documentation; where these occurs clan control is in operation (Kirsch et al., 2010). ISD teams can define and develop a common set of values and norms through interaction and communication between team members (Kirsch, 2004). In ASD regular participation of the customer in the team and collaboration with the team allows them to influence the remainder of the team on their priorities and needs, and ensure that they develop common goals and interests (Hoda et al., 2011). ASD teams may also develop similar values, which can be re-inforced by the team in a number of ways. For example, ensuring that the team display an '*honest*' Product Backlog and the Scrum Board, that '*knowledge sharing*' and '*learning*' takes place by rotating roles and tasks (Schwaber and Beedle, 2002, p119). The factors discussed in the findings chapter for this resultant characteristic are detailed in Table 2.5.4-6.

Factor	
F1	Members are influenced by shared norms
F2	Members are influenced by shared values and beliefs

Table 2.5.4-6 Factors for influenced by shared norms, values and beliefs

- (g) The clan *evaluates members and rewards or sanctions* them based on whether members act in *accordance with the clan's norms, values and goals* (Das and Teng, 1998; Kirsch, 1996; Ouchi, 1980). In an organisational setting formal reward systems have a strong influence on how individuals and teams function (Cohen and Bailey, 1997). They evaluate teams or individuals against pre-defined criteria and information provided by their supervisor, and the feedback received from the evaluator(s) ascertains whether rewards or sanctions are implemented (Flamholtz et al., 1985; Ilgen, Fisher and Taylor, 1979). They are an incentive for individuals to behave in a particular way, to achieve the goals of the organisation, because there is a desire to attain a reward (Flamholtz et al., 1985). These rewards are known as extrinsic rewards as they are tangible and may be in the form of bonuses, promotion, or praise whereas punishment can be in the form of criticism, rewards being given to others, demotion or loss of a job (Merchant and Otley, 2007; Otley, 1999). In contrast, intrinsic rewards are

intangible rewards that are administered to individuals outside of the formal rewards system (Flamholtz et al., 1985) and are a key component of a clan control system with intrinsic rewards such as a feeling of satisfaction or a feeling of accomplishment on the completion of a specific task, or increased responsibility or belonging a strong motivator for individuals (Herzberg, 1968; Merchant and Otley, 2007). As members of a clan share common goals there is no requirement for specific incentives to encourage the achievement of pre-defined goals. Therefore, rewards in a clan tend to be intrinsic, and intangible rather than extrinsic (Kirsch et al., 2002).

Evaluation refers to the assessment of performance (Eisenhardt, 1985), but in clans performance evaluation is subtle and takes place amongst team members in the form of “*reading of signals*”, but it is difficult to translate these signals into something that can be measured (Birnberg and Snodgrass, 1988; Ouchi, 1980). Even though it is not possible to measure outcomes it is possible to reward those who engage in rituals and ceremonies and display the attitudes and values expected by the clan to achieve the desired goals even though it may be difficult to determine if individuals are actually achieving them (Ouchi, 1979). Where clan control exists members of the clan evaluate each other and rewards are based on whether members have acted according to the agreed norms, values and goals of the clan; for example, a member may be rewarded by recognition from the clan (Choudhury and Sabherwal 2003; Kirsch et al., 2002; Kirsch, 2004). Sanctions for non-conformance to the accepted norms or values of a clan are typically informal and may exist in the form of peer pressure, criticism, sarcasm, a disapproving look, or may even be as severe as exclusion, or distancing oneself from the member who exhibits deviant behaviour (Kirsch, 2004; Westphal and Khanna, 2003). In ASD teams the frequent interaction and communication between members can be used as a means to monitor and evaluate performance of team members and provide feedback to each other as it becomes clear very quickly if a team member is not performing as expected (Asproni, 2004). The factors discussed in the findings chapter for this resultant characteristic are detailed in Table 2.5.4-7.

Factor
F1 Members evaluate each other in accordance with the team's norms, values and goals
F2 Members reward each other for conformance to the team's norms, values and goals
F3 Members sanction each other for non-conformance to the team's norms, values and goals

Table 2.5.4-7 Factors for team members evaluating, rewarding or sanctioning

The literature which underpins the resultant characteristics of clan control is summarised in Table 2.5.4-8. The table differentiates between the ISD/ASD literature and non-ISD/ASD literature.

Resultant Characteristic of Clan Control	References	ISD/ASD References
Members exhibit a strong sense of identity with the clan	(Carmeli et al., 2005; Eckel and Grossman, 2005; Kanter, 1968; Lee, 1971; Rousseau, 1998; Turner and Makhija, 2006; Van Der Vegt and Bunderson, 2005; Wenzel, 2004)	(Braithwaite and Joyce, 2005; Kirsch, 1996; Marks and Lockyer, 2005; Whitworth and Biddle, 2007b)
Members exhibit a strong sense of commitment to the clan	(Bishop et al., 2000; Buchanan, 1974; Kanter, 1968; Larson and LaFasto, 1989; Macintosh, 1994; Mowday et al., 1982; Ouchi, 1979)	(Asproni, 2004; Boehm and Turner, 2005; Chow and Cao, 2008; Hoda et al., 2011; Misra et al., 2009; Nerur et al., 2005)
Members develop a strong sense of loyalty to the clan	(Alvesson and Lindkvist, 1993; Buchanan, 1974; Macintosh, 1994, p38; Ouchi, 1980)	(Mak and Sockel, 2001; Whitworth and Biddle, 2007b)
Members develop common interests and goals, which evolve as tasks progress	(Alvesson and Lindkvist, 1993; Ouchi, 1979; Turner and Makhija, 2006)	(Gopal and Gosain, 2010; Harris et al., 2009; Kirsch and Cummings, 1996; Kirsch, 2004; Maruping et al., 2009; Walz et al., 1993)
Rituals and ceremonies identify and reinforce acceptable behaviours with members attempting to be 'regular' members of the clan by adhering to agreed behaviours	(Hegtvedt, 2005; Kanter, 1968; Macintosh, 1994; Ouchi, 1979; Ouchi, 1980)	(Kirsch, 1996; Kirsch, 1997; Kirsch et al., 2010)
Members are influenced by the shared norms, values and beliefs that exist in the clan	(Birnberg and Snodgrass, 1988; Jaworski, 1988; Kanter, 1968; Lebas and Weigenstein, 1986; Macintosh, 1994; Ouchi, 1980; Wilkins and Ouchi, 1983)	(Hoda et al., 2011; Kirsch, 1997; 2004; Kirsch et al., 2002; Kirsch et al., 2010)

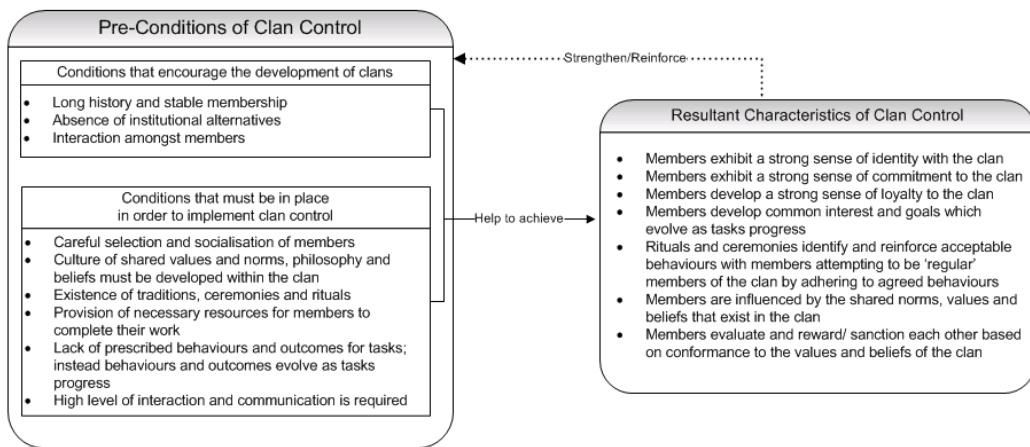
Resultant Characteristic of Clan Control	References	ISD/ASD References
Members evaluate and reward or sanction each other based on conformance to the norms, values and goals of the clan	(Birnberg and Snodgrass, 1988; Cohen and Bailey, 1997; Das and Teng, 2001; Flamholtz et al., 1985; Herzberg, 1968; Merchant and Otley, 2007; Ilgen et al., 1979; Otley, 1999; Ouchi, 1980; Westphal and Khanna, 2003)	(Choudhury and Sabherwal 2003; Kirsch, 1996; Kirsch, 2004; Kirsch et al., 2002)

Table 2.5.4-8 Resultant characteristics of clan control

2.5.5 Conceptual Framework

The exercise of any form of control means that the behaviours of individuals are influenced in a number of different ways; for example, by actions, systems (e.g. formal systems, belief systems), or by other individuals. As shown in Figure 2.5.5-1 (p.44) the development and implementation of clan control can be determined by a number of pre-conditions. All of these conditions help to motivate individuals to behave in particular ways, which results in the presence of clan control. The presence of clan control is evident in a team or an organisation where the resultant characteristics identified in Figure 2.5.5-1 exist; for example, shared values and norms influence individuals to behave in a particular manner. Each of these factors further strengthen and re-inforce the development and implementation of clan control.

Each of the areas identified in the section 2.5.4 are studied in detail in many different domains. It is argued that this research synthesised the literature to develop and present a comprehensive framework for the pre-conditions and resultant characteristics of clan control within which clan control can be assessed and evaluated. This framework is used to examine clan control in the context of ASD teams (section 3.5, p.74).

**Figure 2.5.5-1 Conceptual Framework: Clan Control**

2.6 Conclusion

This chapter first introduced control theory and established its applicability for this study. It then examined control at organisational level, the focus of many early studies on control, which later progressed to studies of control at team level. A number of control frameworks were presented, which revealed the different control modes that exist in the literature. These were generally divided into two broad categories of control: formal control and informal control. While the importance of both modes of control were recognised, research has predominantly studied formal controls as they tend to be easier to understand and operationalise. However, some researchers have recognised the need to study informal controls, such as clan control, which focus on people and the environment in which they work. Clan control is one of the lesser studied control modes with many researchers acknowledging that it is difficult to conceptualise and operationalise. This mode of control is the focus for this study.

Much of the existing research on clan control used Ouchi's (1979; 1980) control framework as a foundation where clan control is the control of choice in situations where it is difficult to measure outcomes and behaviours. An examination of the literature was conducted to determine existing frameworks for clan control. Different researchers discussed various aspects of clan control, some of which overlapped, such as how clan control may develop, various clan control mechanisms, and the

resultant characteristics of clan control. A series of papers by Kirsch consolidated various concepts and characteristics of clan control, which were drawn from the existing literature. These were used as the initial basis for developing a conceptual framework for clan control. However, further inspection of the literature revealed a number of other pre-conditions of clan control, an additional resultant characteristic of clan control and also conditions for clans to develop. This chapter concludes by synthesising the literature into a single framework for clan control, which may be used by researchers in future studies of clan control.

The framework comprises a set of inter-related elements: pre-conditions that encourage the development of clan control, pre-conditions that must be in place in order to implement clan control, which help to achieve various resultant characteristics of clan control. Each element of the framework was discussed and a number of factors were identified for the resultant characteristics of clan control. These were used to guide the development of the research instrument and were also used as a lens to analyse the empirical data gathered. This conceptual framework is brought forward to the next Chapter where it is further examined in the context of ASD teams.

3 Agile Software Development

3.0 Introduction

There are a number of objectives for this chapter. The first is to introduce agile methodologies and where it fits in the evolution of software development methodologies. This is followed by a summary of the debate in the literature between proponents and opposers of agile methodologies (section 3.1). The concepts of agile methodologies, the principles and practices that they adhere to, and how they differ from the traditional plan-driven approach to software development from a management and control perspective are examined in section 3.2. This section also provides detail on two specific agile methodologies namely; eXtreme Programming (XP) and Scrum. Section 3.3 introduces the three agile practices which are the focus of this study namely; the sprint planning meeting (section 3.3.2), the daily scrum meeting (section 3.3.3), and the sprint review and retrospective meeting (section 3.3.4). Section 3.4 examines the literature on control in ASD teams with a more in-depth analysis of the literature on clan control in ASD teams (section 3.4.2) and an identification of the knowledge gap in this area, although earlier reference was made to ASD in section 2.5.4 (on p.31). This chapter culminates with the presentation of the conceptual framework used for this study (section 3.5).

3.1 Evolution of Software Development Methodologies

The process of developing software development has evolved from ad-hoc, to plan-driven to the more recent flexible, agile approach to software development. There has been much debate in the literature on the most appropriate methodology or process for developing software (Ågerfalk and Fitzgerald, 2006) with some advocating the superiority of agile methodologies over a plan-driven approach (Beck and Andres, 2005; Beck and Boehm, 2003) and vice versa (Parnas, 2006). Others recognise that each methodology has its own merits and is more appropriate for certain types of projects (Abrahamsson, Warsta, Siponen and Ronkainen, 2003; Boehm, 2002; Cockburn and Highsmith, 2001; Harris et al., 2009), or that the methodology must be tailored to suit the needs of an organisation (Conboy and

Fitzgerald, 2010; Fitzgerald, Hartnett and Conboy, 2006; Fitzgerald, Russo and O'Kane, 2003; Karlsson and Agerfalk, 2009).

In the early days, software development was very costly, took a long time to complete and the resulting solution didn't always work very well (Lycett and Paul, 1999). Consequently, a structured, sequential process for software development was presented by Royce (1970) that he believed would remove the risks associated with ad-hoc software development resulting in the delivery of a product that the customer requires. This introduced an engineering approach to software development (Nerur et al., 2005) where each step in the process brought the software product closer to completion in a linear, sequential process from planning, analysis and design in the early stages to testing and implementation in the later stages (Royce, 1970) as shown in Figure 3.1-1.

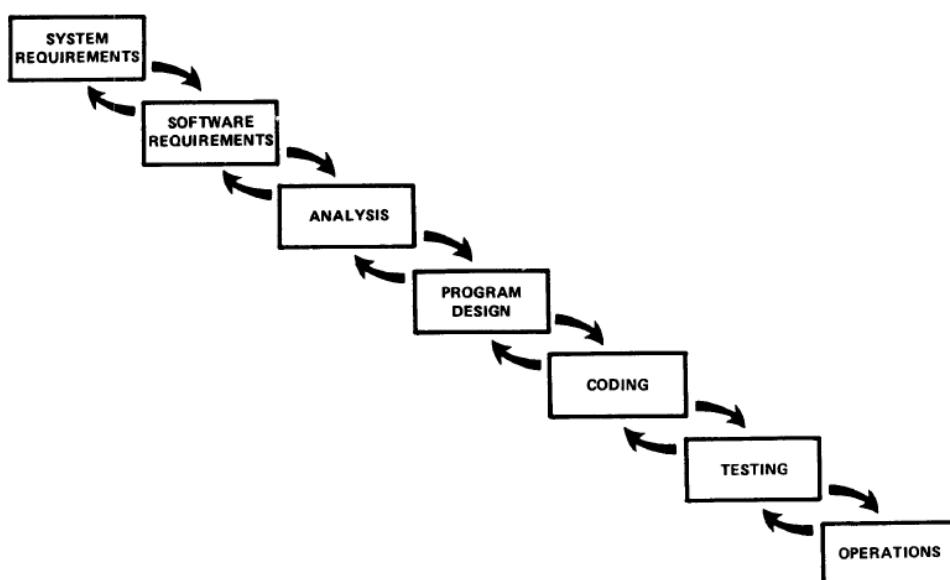


Figure 3.1-1 Waterfall Model Royce (1970)

This approach required the clear definition of tasks, and the assignment of each task to a specific role (Nerur et al., 2005). In addition to the delivery of the software product each step in the process required copious amounts of documentation as Royce (1970) believed that it was not possible to manage a project without having

sufficient documentation in place that detailed what the project did and recorded decisions. Communication was formalised by the adherence to this process and by the creation of numerous documents (Jurison, 1999; Nerur et al., 2005). This sequential process later became known as the “Waterfall” methodology, or a plan-driven approach to software development and has been widely used in industry for many years.

This methodology is very desirable in many software projects, particularly safety-critical projects where controls exist to manage budgets, requirements, schedule etc. and developers are limited in what they can modify within the constraints of the project (Harris et al., 2009). While this methodology removed the process variability that was experienced historically when software development was completed in an ad-hoc fashion (Highsmith and Cockburn, 2001) it has been said that this approach to software development limited the capability of a team to adjust their processes to cater for changing requirements (Nerur et al., 2005) and it also restricted the ability of teams to respond quickly to changing business requirements. This resulted in the development of what is now known as agile methodologies in the late 1990’s (Highsmith and Cockburn, 2001), although the emergence of many agile concepts can be traced back to the early 1980’s in a move away from the traditional software development life-cycle (McCracken and Jackson, 1982) to the concept of prototyping (Connell and Brice, 1984), greater customer involvement (Mahmood, 1987), the use of 4GL tools (Gremillion and Pyburn, 1983), and the spiral model (Boehm, 1988)

Agile methodologies were developed by practitioners in response to the need to find a better, more flexible approach to software development that could address the changing requirements of businesses who needed to deliver software quickly to the market and could overcome the limitations of the plan-driven approach to software development (Bose, 2008; Williams and Cockburn, 2003). Several meetings were held between 17 practitioners, who are now known as the key proponents of agile methodologies, as they sought to define a new, alternative software development methodology (Highsmith and Cockburn, 2001). They realised they were trying to achieve the same goal – to deliver high quality software and have satisfied customers (Williams and Cockburn, 2003). Following several discussions this group of individuals finally agreed on the term ‘agile’ for their collective group

of methodologies, which originated in manufacturing. Each practitioner developed their own version of an agile methodology, but each agile methodology shared the same fundamental values and principles created by these 17 practitioners defined by what is now known as the Agile Manifesto (AgileAlliance, 2001). The Agile Manifesto provides a common framework for all agile methodologies (Cohn and Ford, 2003) by specifying particular values and principles. The values of agile methodologies are as follows (Table 3.1-1):

Agile Values
<i>Individuals and interactions</i> over processes and tools
<i>Working software</i> over comprehensive documentation
<i>Customer collaboration</i> over contract negotiation
<i>Responding to change</i> over following a plan

Table 3.1-1 Values of Agile Methodologies

While each of these values are considered important, agile methodologies place more value in the items on the left than those on the right. Firstly, they place great emphasis on the people involved in a software project and their interactions with each other, which can result in a good working environment and working relationships and improve team spirit (Abrahamsson, Salo, Ronkainen and Warsta, 2002). Secondly, the focus is on the frequent delivery of working software, which is then followed by an emphasis on the interaction and relationship between the customer and the team rather than on formal contracts that specify the requirements for the project (Abrahamsson et al., 2002; Highsmith and Cockburn, 2001). Finally, both the team and the customer should be provided with the capability of responding to change and have the authority to make any necessary changes (Abrahamsson et al., 2002).

In addition to these values 12 principles were also defined (Table 3.1-2), which are part of the Agile Manifesto. They are principles rather than rules intended to guide ISD teams and managers (Williams, 2007).

Agile Principle	
P 1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
P 2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
P 3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
P 4	Business people and developers must work together daily throughout the project.
P 5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
P 6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
P 7	Working software is the primary measure of progress.
P 8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
P 9	Continuous attention to technical excellence and good design enhances agility.
P 10	Simplicity--the art of maximizing the amount of work not done--is essential.
P 11	The best architectures, requirements, and designs emerge from self-organizing teams.
P 12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Table 3.1-2 Twelve Principles of Agile Methodologies

Even though agile methodologies bring many benefits to the development of software they are also criticised by many, but they have a strong following with organisations increasingly adopting agile (Laanti, Salo and Abrahamsson, 2011; Lee and Xia, 2010; Sliger and Broderick, 2008, p10; VersionOne, 2009).

3.2 Agile Methodologies

Agile methodologies is an umbrella term for a group of distinct methodologies, all of which share common values and principles as indicated in section 3.1. No single agile methodology exists. Some examples are eXtreme programming (XP) (Beck and Andres, 2005), Scrum (Schwaber and Beedle, 2002), Crystal methods (Cockburn, 2004) and Feature Driven Development (Palmer and Felsing, 2002). Regardless of the name of an agile methodology, each agile methodology embraces change and is considered most suitable for projects that have high uncertainty or where the requirements change rapidly (Beck and Andres, 2005; Cohn and Ford, 2003). Their objective is to design and deliver high quality software quickly to the

customer (Ceschi et al., 2005; Highsmith and Cockburn, 2001) with an emphasis on productivity rather than how the team conform to a process (Ågerfalk and Fitzgerald, 2006).

Software is developed in increments, called sprints (also called iterations), with each sprint producing a small chunk of work that is delivered to and reviewed by the customer, who provides feedback, which can include modifications to the requirements (Ceschi et al., 2005; Rising and Janoff, 2000). These sprints can be as short as two or three weeks, and should be no more than six weeks (Highsmith and Cockburn, 2001). They promote the development of software at a sustainable pace (Williams, 2003) with the team continuously making progress even if the requirements are unstable (Rising and Janoff, 2000). The customer is an integral part of an agile methodology and is encouraged to be actively involved in the project, even as frequently as daily, for the duration of the project (Ceschi et al., 2005; Highsmith and Cockburn, 2001; Nerur et al., 2005). Their input and feedback is imperative for the success of a project with the customer able to delay decisions until it is absolutely necessary (Bose, 2008). Their participation also provides the team with the opportunity to build a strong relationship with the customer and to really understand the customer requirements, which should result in more satisfied customers (Ceschi et al., 2005; Williams, 2003).

In ASD projects tasks and their progress are visible across the team; team communication improves; and a relationship is built with the customer who begins to place great trust and great faith in the team as they see the progression of their product until it is finally delivered (Rising and Janoff, 2000). While documentation is considered important, less importance is attached to it than in a plan-driven approach to software development (Bose, 2008). A minimalist approach to documentation is taken with only relevant documentation created, which ensures that the documentation reflects the current state of the system (Highsmith and Cockburn, 2001; Nerur and Balijepally, 2007).

	Plan-Driven	Agile
Fundamental Assumptions	Systems are fully specifiable, predictable and can be built through meticulous and extensive planning	High-quality adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change
Control	Process-centric	People-centric
Management Style	Command-and-control	Leadership-and-collaboration
Knowledge Management	Explicit	Tacit
Role Assignment	Individual – favours specialisation	Self-organising teams – encourages role interchangeability
Team size	No restriction	Small, typically < 10 people
Team Input	Team told what to do	Team make the final decision
Communication	Formal and conducted when required	Informal, regular collaboration
Customers Role	Important	Critical
Customer Availability	Availability varies	Easily accessible
Project Cycle	Guided by tasks or activities	Guided by product features
Development Model	Lifecycle model following a linear, sequential process (Waterfall, Spiral, or some variation)	The evolutionary-delivery model
Desired Organisational Form/Structure	Mechanistic (bureaucratic or formalization)	Organic (flexible and participative encouraging cooperative and social action)
Technology	No restriction	Favours object-oriented technology
Documentation	Mandatory	Created as required

Table 3.2-1 Differences between plan-driven and agile methodologies (Nerur et al., 2005; Schuh, 2004, p8)

Agile methodologies have become an accepted choice of methodology in the ISD community in recent years with 84% of respondents to an annual survey conducted by Version One (2009), an organisation that specialises in the agile project planning and management, indicating that they use agile development practices to some extent. Research in ASD has also grown in recent years (Abrahamsson, Conboy and Wang, 2009; Conboy, 2009; McEvoy and Butler, 2009), yet further research is required in many different aspects of ASD. Practitioner focused research such as that produced by Forrester and Gartner regularly report on many different aspects of agile

methodologies and ASD. A review of the literature identifies several differences between the plan-driven approach to software development and agile methodologies (Table 3.2-1).

From a management and control perspective the most notable changes between plan-driven and agile methodologies relate to the movement from a command-control structure to a collaborative approach and the change in the organisation of the team, who are allowed to self-organise, decide how the work is done and delegate tasks to each other (Nerur and Balijepally, 2007; Nerur et al., 2005; Schuh, 2004). Consequently, managers must trust the team and individuals within the team to use their knowledge and skills effectively (Cockburn and Highsmith, 2001). The teams are empowered to make decisions, which are traditionally the responsibility of the project manager or other designated decision-makers (Cockburn and Highsmith, 2001). This gives the team control over their work; they are responsible for managing their own work within the confines of each iteration and they are fully accountable for the delivery of their project (Cockburn and Highsmith, 2001; Sliger and Broderick, 2008). They focus on people and rotate roles to increase the skills, competencies and diversity of the team (Cockburn and Highsmith, 2001; Nerur et al., 2005). Many software development teams consist of individuals who do not necessarily know each other very well and communicate only when required, which is often done using electronic means e.g. email (Williams, 2003). This contrasts with ASD teams where face-to-face communication is considered essential and reduces the amount of time and effort to obtain information or receive feedback (Bose, 2008; Cockburn, 2007). Therefore, the ability of team members to communicate is extremely important as they communicate and collaborate extensively with each other and with various stakeholders (Cockburn and Highsmith, 2001; Nerur et al., 2005). The participation of the customer is critical in an agile project whose role is to define and prioritise requirements and provide continuous feedback to the team, but the customer selected must be committed to the project, be knowledgeable and empowered to make decisions (Boehm, 2002). Finally, there is less emphasis on formal documentation in ASD projects, although this can cause problems later where tacit knowledge is not documented (Boehm, 2002).

XP and Scrum are two of the better known agile methodologies and are considered two of most commonly used methodologies with practitioners. This is evident in the findings of a study conducted in 2009 of 2,750 organisations, which found that 74% of organisations used either Scrum, XP or a hybrid of both as their chosen agile methodology (VersionOne, 2009). This research studied teams that implemented either of these two agile methodologies. Further detail on these methodologies are detailed in section 3.2.1 (p.54) and section 3.2.2 (p.59). These two methodologies complement each other well with XP providing support for technical aspects and Scrum providing support from a management perspective (Fitzgerald et al., 2006). However, few empirical studies focus on specific agile methodologies. In a systematic literature review conducted by Dybå and Dingsøyr (2008) on empirical studies of agile software development conducted up to and including 2005 twenty-five of the thirty-six empirical studies on agile methodologies identified focused on XP, whereas only one study investigated Scrum. There is clear evidence of the need for further empirical research on each agile methodology. More recent studies focus specifically on XP and Scrum, or a combination of both, but these predominantly use a single case study as the basis for their research (Atlas, 2009; Augustine et al., 2005; Dahlberg, Ruiz and Olsson, 2006; Fitzgerald et al., 2006; Layman, Williams and Cunningham, 2006; Moe, Dingsøyr and Dybå, 2010; Sheth, 2009; Sutherland et al., 2007; Vlaanderen, Jansen, Brinkkemper and Jaspers, 2011) with a smaller number using multiple case studies (Karlsson and Agerfalk, 2009; Pino, Pedreira, GarcÃa, Luaces and Piattini, 2010; Tolfo and Wazlawick, 2008). While case study appears to be the preferred approach the study of multiple cases is merited, which is the approach selected for this study and is discussed futher in section 4.4.3 on p.93.

3.2.1 Extreme Programming (XP)

Extreme Programming (XP) (Figure 3.2.1-1), which originated with Kent Beck in 1996, is an agile methodology that “*focuses on the application of programming techniques, clear communication and teamwork*” (Beck and Andres, 2005, p2). It is widely considered the starting point for many other agile methodologies (Abrahamsson et al., 2002). It is called extreme because “*it carries to an extreme degree some of the practices already used in traditional software development*” (Angioni, Carboni, Pinna, Sanna, Serra and Soro, 2006).

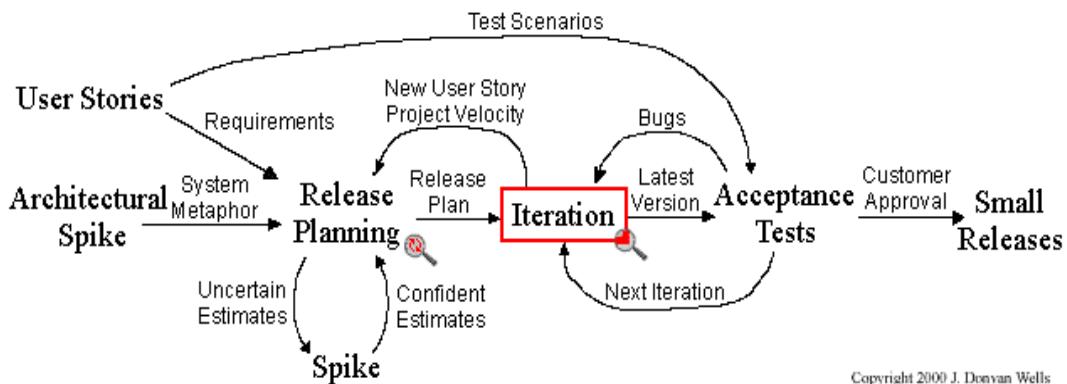


Figure 3.2.1-1 Extreme Programming Process

The methodology recommends first writing user stories that document the features for the next release of the software system. Where necessary a ‘spike’ is conducted that allows a team member an opportunity to explore potential solutions for a difficult technical or design problem in order to provide a more accurate estimate. All this information is required for the release planning meeting, which requires the participation of both the team and the customer. The customer prioritises user stories, estimates are determined for each user story by the team and the release plan is agreed. The release plan divides the development into a number of iterations (timebox), which may be as short as two weeks. The length of each iteration remains constant through the project with the customer selecting the user stories (scope) needed to fill the iteration length at the iteration planning meeting (section 3.3.2 on p.65), which is held at the start of each iteration. Testing is completed within each iteration. When the last iteration is completed the software is ready for release to the customer (Abrahamsson et al., 2002; ExtremeProgramming, 2009b; Williams, 2007).

An XP team consists of several roles where each role interacts and works with each other, although roles on a mature XP team are not fixed as long as each person contributes to the team (Beck and Andres, 2005, p82). These roles include the manager, the coach, the tracker, the programmer, the tester and the customer. The manager facilitates the communication within the team and with any external stakeholders. They also coordinate resources, keep track of progress and manage

problems. The coach is typically a developer, teaches team members about the process and monitors team members to ensure they follow the process. The tracker, is also typically a programmer and keeps track of the progress of user stories and testing. The programmer estimates times for user stories, writes and tests the code, with the tester helping the customer to write and develop tests and finally, the customer defines and writes the user stories, selects and prioritises the user stories for each release and provides feedback to the team (Abrahamsson et al., 2002; Beck and Andres, 2005; Williams, 2007).

Like other agile methodologies XP requires teams to deliver working software in a short timeframe i.e. weeks; to be flexible to cater for adjustments to the software product; to keep the design simple; to improve design; and to test early and often with the aim of reducing defects and achieving high quality software (Beck and Andres, 2005, p2; Highsmith and Cockburn, 2001). XP embraces 5 values, communication, simplicity, feedback, respect and courage (Table 3.2.1-1), although these values are not the only values that a team may have (Beck, 2000).

Value	Description
Communication	Communication is critical in XP to provide feedback, communicate difficulties and solve problems as quickly as possible. It also helps to create a cooperative team.
Simplicity	XP recommends that a system is developed as simply as possible and to avoid unnecessary complexity such as coding for unstated requirements
Feedback	XP recommends continuous feedback, which should occur as quickly as possible; for example, feedback on technical decisions, customer requirements, or errors.
Courage	Team members must face and voice any fears that they have in an attempt to address them as quickly as possible; for example, requesting help to resolve a problem, or expressing dissatisfaction with the quality of work produced by another team member, or resisting pressure to make unrealistic commitments
Respect	All team members must care about each other and what they are doing

Table 3.2.1-1 Key Values of eXtreme Programming (Beck, 2000)

The initial version of XP defined 12 developer-focused technical practices (Table 3.2.1-2), which were based on these values. While these practices were not new and were not the only practices that could be used by ISD teams, they were combined to

form the practices for XP (Abrahamsson et al., 2003; Beck, 2000). It is important to remember though that these practices are not a set of rules that must be implemented in all teams that use XP, rather they detail a common way for teams to work with teams choosing which practices are most appropriate for their unique situation (Beck and Andres, 2005; Highsmith and Cockburn, 2001).

Practice	Description
The planning game	A quick determination of the scope of the next software release, based on a combination of business priorities and technical estimates. It is accepted that this plan will probably change
Small releases	Put a simple system into production quickly, and then release new versions on a very short cycle.
Metaphor	Guide all development with a simple shared story of how the whole system works.
Simple design	The system should be designed as simply as possible at any given moment in time.
Testing	Programmers continually write tests, which must be run flawlessly for development to proceed. Customers write function tests to demonstrate the features implemented.
Refactoring	Programmers restructure the system, without removing functionality, to improve non-functional aspects (e.g. duplication of code, simplicity, flexibility).
Pair-programming	All production code is written by two programmers at one machine.
Collective ownership	Anyone can change any code anywhere in the system at any time.
Continuous integration	Integrate and build the system every time a task is completed – this may be many times per day.
40-H week	Work no more than 40 hours per week as a rule
On-site customers	Include an actual user on the team, available full-time to answer questions.
Coding standards	Adherence to coding rules that emphasise communication via program code.

Table 3.2.1-2 Key Practices of eXtreme Programming (Beck, 2000)

However, based on feedback from industry Beck and Andres (2005) later revised the practices of XP to include 13 primary practices and 11 corollary practices (Table 3.2.1-3).

Technical Practices	Description	Corollary Practices	Description
Sit together	The whole team sits together in one open space	Real customer involvement	Make the customer part of the team and ensure they are empowered to make decisions about requirements and their priority
Whole team	Ensure the team includes people with the necessary skills, perspectives and expertise to complete the project	Incremental deployment	Deploy functionality to a production environment incrementally
Informative workspace	The workspace should inform an observer of how the project is going – use the wall space to display charts	Team continuity	Keep effective teams together. Value their relationships and what they accomplish together
Energised work	Do not work excessive overtime for long periods of time	Shrinking team	As the team becomes more experienced gradually reduce the size, but keep the workload constant
Pair programming	Two programmers work together at one machine on a single piece of code	Root cause analysis	Eliminate each defect as it is found, identify its cause and remove it.
Stories	Short description of user functionality	Shared code	Share code amongst team members. Allow all team members to modify any piece of code
Weekly cycle	Meet each week and plan the work for that week	Code and test	Maintain only the code and the test as permanent artifacts. Rely on social mechanisms to keep alive the important history of the project.
Quarterly cycle	Once a quarter reflect as a team and plan at a high level the work for the next quarter	Single code base	Keep a single code base. Avoid multiple versions of the code base.
Slack	Include some low-priority tasks that can be dropped if needed	Daily deployment	Put the new software into production every night
10 minute build	Be able to build the whole system and run all tests in 10 minutes	Negotiated scope contract	Fix the time, cost and quality of a project, but allow room to negotiate the scope of the project on an on-going basis
Continuous integration	Integrate code and test changes several times a day.	Pay per use	Charge for every time the system is used by a user
Test-first programming	Write tests before coding		
Incremental design	Develop the design as the project progresses		

Table 3.2.1-3 Revised Practices of eXtreme Programming (Beck and Andres, 2005)

The primary practices were intended to be independent of each other and their use depended on the environment within which a team functions, whereas the corollary practices should only be considered once some of the primary practices are successfully introduced (Beck and Andres, 2005). Even though it is not one of the official XP practices, all teams also have short daily stand-up meetings (section 3.3.3 on p.66), where team members tell the team what s/he accomplished the previous day, what s/he plans to do today, and any obstacles or difficulties that s/he is experiencing (Williams, 2007).

3.2.2 Scrum

Scrum is an agile methodology that enables teams to work together effectively to deliver software incrementally to meets the needs of customers (Schwaber and Beedle, 2002). It emphasises project management techniques rather than the engineering practices of XP (Schwaber, Leganza and D'Silva, 2007). The term was first used in Japan in the 1980's by Takeuchi and Nonaka (1986) where it described a self-organising product development process that was similar to a strategy used in rugby for "*getting an out-of-play ball back into play*" with teamwork (Schwaber and Beedle, 2002, p1). Jeff Sutherland (2004) and his colleagues claim to have first used Scrum in software development in 1993, followed closely by Ken Schwaber. Scrum embraces five values: commitment, focus, openness, respect and courage (Table 3.2.1-1) all of which emerge as team members participate in Scrum.

Value	Description
Commitment	Teams must be willing to commit to a goal. Scrum provides teams with the authority they need to meet their commitments
Focus	Scrum recommends focusing all efforts and skills on doing the work that each individual has committed to doing.
Openness	Scrum keeps everything about a project visible to everyone.
Respect	Individuals are shaped by their background and their experiences. It is important to respect the different people who comprise a Scrum team,
Courage	Have the courage to commit, to act, to be open and to expect respect

Table 3.2.2-1 Key Values of Scrum (Schwaber and Beedle, 2002, p147).

Scrum was designed to allow developers to self-organise into high-performance teams where they can define their goal for each sprint, and are given the authority and autonomy and responsibility on how best to meet the goal (Sutherland et al., 2007; Williams, 2007). It focuses on using small teams of less than ten people to work on projects that are developed in a series of short timeframes called sprints (Cohn and Ford, 2003; Rising and Janoff, 2000) and defines three main practices summarised in Table 3.2.2-2. Further detail on each of these agile practices is provided in section 3.3 (p.63).

Agile Practice	Description
Sprint Planning	The sprint planning session is a meeting that takes place at the start of each sprint where the team collectively define and plan tasks that must be completed during the next sprint
Daily Scrum	The daily scrum is a short daily status team meeting lasting a maximum of 10-15 minutes typically conducted at the same time each day. The meeting is conducted with team members standing up. During the meeting team members explain briefly what they accomplished since the previous meeting, what will be completed by the next meeting and indicate any impediments that may prevent them from completing their current tasks
Sprint Review and Retrospective	A sprint review and retrospective is a meeting that is held at the end of each sprint where the project team demonstrates the software completed to the customer and to management to obtain feedback. They then reflect on what went well in the sprint, what did not, and what could be improved for future sprints

Table 3.2.2-2 Key Practices of Scrum (Elssamadisy, 2007; Schwaber and Beedle, 2002).

In Scrum there are three main roles, the Scrum Master, the Product Owner and the Team, although other roles also exist (Schwaber and Beedle, 2002). The **Scrum Master** is the management role in a Scrum team and they are responsible for ensuring the scrum values and practices are implemented (Schwaber and Beedle, 2002, p31). Their primary function is to remove any obstacles that the team may face (Cockburn, 2007, p263; Schwaber and Beedle, 2002, p45). The **Product Owner** represents the customer; manages and prioritises the customer requirements; is empowered to make decisions on behalf of the customer; and manages the product vision and the return on investment. The final role is the **Scrum Team** who organise themselves and commit to do the work (Schwaber and Beedle, 2002). This contrasts with XP where a number of roles exists, but they are not fixed rather, individuals can

freely share and rotate roles as they see fit with individuals potentially gaining experience in a variety of roles and contributing where they can (Beck and Andres, 2005, p82).

Scrum consists of three phases: the pre-game phase, the game phase, and the post-game phase (Figure 3.2.2-1).

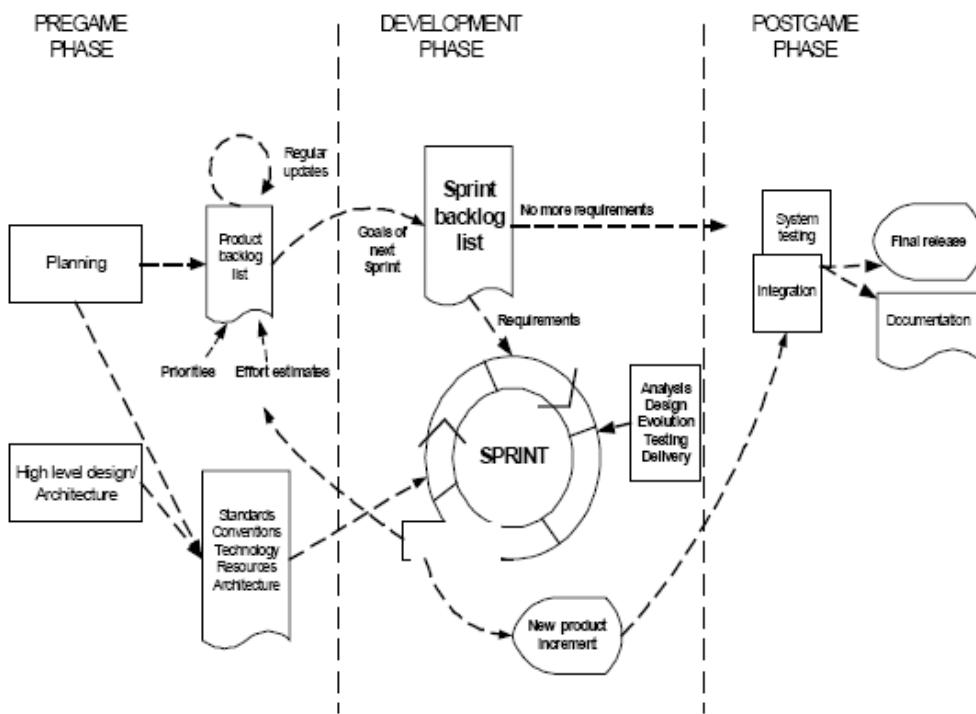


Figure 3.2.2-1 The Scrum Process (Abrahamsson et al., 2002)

The **Pre-Game** phase consists of two elements. The first element is the planning phase. This is where the Product Backlog is created that contains a prioritised list of product features, technologies, issues and enhancements for the project, which can evolve and change over time as requirements change (Schwaber and Beedle, 2002, p33). The items on the Product Backlog are prioritised by the customer (Product Owner) who is also responsible for maintaining the Product Backlog (Schwaber and Beedle, 2002, p33). Planning also includes determining resources, tools, risks and any other issues applicable to the project (Abrahamsson et al., 2002). The second element is the architecture phase, which includes system architecture modification and high-level architecture design as to how the backlog items will be implemented.

The **Game** phase, also known as the ***Development*** phase is the agile part of the methodology. This phase involves the development of a new release of functionality in short timeframes, called sprints, with constant reference in each sprint to time, requirements, quality and cost (Schwaber and Beedle, 2002). Multiple, iterative development sprints, or cycles, are used to develop the system with each sprint containing some systems analysis, design, development, test and delivery (Abrahamsson et al., 2002; Schwaber and Beedle, 2002). At the start of each sprint the scrum team hold a sprint planning meeting (section 3.3.2 on p.65) where the team meet with the customer (Product Owner) to set the goal for the sprint, prioritise the work and to select the tasks for completion within the sprint, which are detailed on the Sprint backlog (a sub-set of the product backlog) (Cohn and Ford, 2003). The team break down the high-level features selected from the Product Backlog into tasks on the Sprint Backlog that can be estimated, with each task assigned to a team member (Williams, 2007). The Sprint Backlog lists the features and time estimates assigned for completion within the current sprint (Schwaber and Beedle, 2002, p71; Williams, 2007). The number of hours available to complete the tasks for the sprint are graphed and made visible to the team in a burndown chart with the team aspiring to reach zero at the end of each sprint. During the sprint the team hold daily meetings, called daily scrums (section 3.3.3 on p.66) to briefly review progress (Cohn and Ford, 2003). At the end of each sprint a sprint review and retrospective meeting is held where a version of the product is demonstrated to the customer and to management for review and feedback, which can potentially be released to production (Cohn and Ford, 2003). The team further conduct a review where they reflect on the successes or failures of the sprint (section 3.3.4 on p.68) (Schwaber and Beedle, 2002).

The ***Post-game phase*** is the final phase where the focus of the team is on preparation for final release to the customer, including final documentation, pre-release staged testing, and release. All tasks are completed at this stage and no new requirements are permitted (Abrahamsson et al., 2002; Schwaber and Beedle, 2002).

3.3 Agile Practices Investigated

This section first explains what is meant by an agile practice. It then justifies the selection of the chosen agile practices (section 3.3.1), provides the reader with an understanding of each of the three selected agile practices and how they are implemented as prescribed by the methodology as detailed in commonly referenced agile textbooks (sections 3.3.2, 3.3.3, and 3.3.4), and concludes by examining the implementation and tailoring of agile practices in ASD teams (section 3.3.5, p.69).

Each agile methodology details various practices that distinguish it from other agile methodologies, but they each follow the same underlying agile principles (Agile Alliance, 2001). A practice can be described as a “*common way of acting*”, which is accepted by a group of individuals as the “*correct way to do things*” (Hansson, Dittrich, Gustafsson and Zamak, 2006). Agile practices are the things that ASD teams do on a daily basis to “*stay disciplined, foster communication, remain flexible, and deliver software*” (Schuh, 2004, p4). It is not essential for ASD teams to adopt all practices of an agile methodology at once as this may result in too much change for a team, which they may find difficult to handle (Beck and Andres, 2005, p55; Schwaber et al., 2007). However, the practices can improve software project quality if they are implemented correctly (Sfetsos and Stamelos, 2010). Some agile practices are technical and focus on developers, whereas others are management focused, but according to Schuh (2004, p5) each practice will only work successfully if it is implemented and used correctly. Teams may find it easier to gradually introduce the practices over time – as they become familiar with one practice another practice may be introduced (Beck and Andres, 2005, p55). Thus, it is recommended to adopt the practices incrementally to allow teams gain experience with the methodology (Elssamadisy, 2007), or to tailor the methodology to a particular context (Conboy and Fitzgerald, 2010; Fitzgerald et al., 2006). This means that teams can choose to adopt only the agile practices that suit their environment or that work well for them, which can also span several agile methodologies, with many practitioners adopting practices from different agile methodologies to use a hybrid methodology; for example, a mixture of the practices of Scrum and XP (Elssamadisy, 2007; Hansson, Dittrich, Gustafsson and Zarnak, 2006).

3.3.1 Selection of Agile Practices

Three agile practices were selected as the focus of this study: the sprint planning meeting, the daily scrum meeting, and sprint review and retrospective meeting as detailed by the Scrum methodology (Table 3.2.2-2, p.60). XP also uses similar practices, which are fundamentally the same, but have differing names. In XP these practices are called the iteration planning, daily stand-up, and iteration retrospective. To avoid confusion the Scrum terminology for these practices are used from this point onwards. Also, the Scrum term Product Owner will refer to the customer representative on the team. These three agile practices were selected for the following reasons:

- Each agile methodology specifies a collection of practices that define a common way of working. For example, XP and Scrum alone detail 27 different practices (Table 3.2.1-3 on p.58 and Table 3.2.2-2 on p.60) and a study by Abbas, Gravell and Wills (2010) examined 58 different agile practices. As a large number of agile practices exist it would prove difficult to examine each practice in detail in a single study. Therefore, a subset was of practices were selected for this study.
- Teams are not required to implement every practice, rather they are encouraged to select practices that meet their needs (Beck and Andres, 2005, p36) with many teams tailoring a methodology to suit their particular needs (Conboy and Fitzgerald, 2010; Fitzgerald et al., 2006). Therefore, finding teams that had implemented all agile practices as prescribed by a particular methodology could prove difficult.
- The agile practices selected are amongst the more commonly used agile practices by practitioners (VersionOne, 2009) and it was believed that it would be reasonably straightforward to find teams that used these all three practices.
- The conceptual framework presented in Figure 3.5-1 (p.77) contains seven different concepts. There was a trade-off between investigating a smaller number of characteristics or a smaller number of agile practices. However, it was considered more important to have a strong framework and investigate fewer practices, rather than examining more practices in a diluted framework. The

study of other agile practices could be conducted at a future date. See section 7.5, p.285 for details on future research.

- As stated in section 3.3 (p.63), each agile practice has a different focus. Some are more technical and are particular to certain roles within a team. For example, pair programming is a practice used solely by developers, and continuous integration is a practice that may be used by QA and developers, whereas other practices focus on the management and control of the project. The agile practices selected were considered more applicable to clan control than other practices as they complement each other with each of the practices logically related to the management and control of an agile project (Laplante, 2003; Abbas et al., 2010). They also require frequent interaction and communication amongst all team members for the duration of a project, which is an important component of ASD (Schwaber and Beedle, 2002).
- Finally, in order to conduct an in-depth study of clan control where the whole team was considered a clan, it was important to obtain the views of all team members. The three practices selected required the regular participation of all team members. Selecting practices that did not necessitate the participation of every team member would have limited the findings of the study.

3.3.2 Sprint Planning Meeting

The sprint planning meeting occurs at the start of each sprint to plan the deliverables for the sprint and consists of two parts (ExtremeProgramming, 2009c). While many teams start their sprint on a Monday, sprints can start on any day of the week (Beck and Andres, 2005, p46). The first part involves a meeting between the Product Owner, the Scrum Master and the team, which is led by the Product Owner and can last several hours (Schwaber, 2004, p133). The team review the Product Backlog (list of requirements) in conjunction with the Product Owner, who prioritises the tasks and determines which tasks are included in the sprint (Cohn and Ford, 2003; Schwaber and Beedle, 2002). The team can negotiate with the Product Owner on the amount of work they believe can be completed in the sprint with the final list of tasks and the goal for the sprint agreed between the team and the Product Owner, which

then becomes the Sprint Backlog (Rising and Janoff, 2000; Schwaber and Beedle, 2002; Williams, 2003).

The second part of the meeting involves the team where the team start to self-organise, make decisions, collectively estimate times for each task, identify how the work will be done, and select the tasks that they will complete during the sprint in order to meet the sprint goal (Rising and Janoff, 2000; Williams, 2003; Schwaber and Beedle, 2002). This requires cooperation from all team members (Beck and Andres, 2005, p93). Estimation in particular can prove difficult, especially at the outset, with estimates often varying widely, but a collaborative approach to estimation helps to reduce the variability and provide more accurate estimates (Beck and Andres, 2005, p94).

The Product Owner can remain for this part of the meeting to answer any questions the team may have about the Product Backlog, but all other team members are required to attend (Schwaber, 2004, p134; Schwaber and Beedle, 2002, p49). It is important for all team members to participate and contribute ideas, as each opinion is valuable (Beck and Andres, 2005, p93), but it is equally important to listen to each other and discuss the benefits and drawbacks of suggested solutions (Subramaniam, 2006, p23). No new functionality may be added to the sprint backlog during a sprint (Rising and Janoff, 2000).

3.3.3 Daily Scrum Meeting

Both XP and Scrum require the agile team to meet daily, ideally face-to-face, in what is known as the daily scrum meeting (Scrum) or the daily stand-up meeting (XP), which replaces all other status update meetings (ExtremeProgramming, 2009a). The agile team (Product Owner, Scrum Master and the scrum team) meet at the same time each day, in the same location, for approximately 10-15 minutes (Cockburn, 2007; Schwaber and Beedle, 2002), although only the scrum team and the Scrum Master are permitted to speak (Williams, 2007). Team members are expected to arrive on time and the meeting should start as scheduled regardless of whether all team members are present with team members standing to encourage a short meeting (Schwaber and Beedle, 2002; Williams, 2007). The meeting must include any distributed team members who can communicate via a conference phone or video

conferencing (Rising and Janoff, 2000). Any other employees within the organisation can also be present, but they must remain silent during the meeting (Schwaber and Beedle, 2002). The location where the daily scrum is held should be equipped with the tools required by the team e.g. conference phone, whiteboard, pens, scrum board (Schwaber and Beedle, 2002, p43).

The purpose of the daily scrum is to improve communication and knowledge of the project, promote quick decision-making, keep the team focused, communicate progress and obstacles, identify ways to remove any obstacles and finally to build relationships with the customer and each other helping them to feel part of the team (Rising and Janoff, 2000; Schwaber and Beedle, 2002). They can cause shy or introverted individuals to dread the daily meeting as they are forced to speak publicly every day (Laplante, 2003) with each team member speaking in turn to answer the following three questions (Schwaber and Beedle, 2002, p43; Williams, 2003):

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

The meeting is not used to problem-solve or to resolve issues as these are discussed after the meeting with the relevant personnel (Rising and Janoff, 2000; Schwaber and Beedle, 2002, p46). Instead, it is used as a forum to identify obstacles where individuals can seek help or provide help (Williams, 2003). Yet, if the meeting is not used correctly it can become a form of command-and-control for an individual with authority (Laplante, 2003).

During the daily scrum team members publicly make commitments to themselves and to each other to complete tasks and they are accountable for the tasks selected (Williams, 2003). Estimates for tasks may be modified as it becomes obvious that a task may take more or less time than originally expected and the number of hours of estimated remaining work is updated (Schwaber and Beedle, 2002, p49). At all times the team strive to meet the sprint goal, but it is quickly highlighted when tasks will not reach their deadline as tasks are visible to all team members (Rising and Janoff, 2000). When this occurs the Scrum Master must meet with the Product Owner and

the scrum team to either remove or reduce the scope of functionality from the sprint that will allow the team to meet the goal of the sprint (Schwaber and Beedle, 2002, p50). It also becomes obvious very quickly where there are difficulties within the team e.g. personal conflicts, individual under pressure (Schwaber and Beedle, 2002). Daily scrums are intended to be supportive, but personality clashes can be quickly highlighted and they may also be used as forums to criticise individuals, which can cause anguish for the affected individuals (Laplante, 2003). It is recommended that instead of criticising individuals, team members should focus on the idea or problem and clearly communicate their reasons for concern (Subramaniam, 2006, p19).

3.3.4 Sprint Review and Retrospective Meeting

On the last day of each sprint a review and retrospective meeting is held with all team members, the Product Owner, management, and any other interested stakeholders, which is coordinated by the Scrum Master (Schwaber and Beedle, 2002, p55). The meeting is generally timeboxed to a defined number hours, although this can vary from team to team, and consists of two parts (Schwaber, 2004, p137). The first part (the review) involves an overview of the sprint goal with a synopsis of any difficulties that the team encountered or any limitations in the software (Schwaber and Beedle, 2002, p55). This is followed by a demonstration of the software that has been built in the current sprint (Rising and Janoff, 2000). Functionality that is not complete cannot be demonstrated (Schwaber, 2004, p137). Management and the Product Owner review the software developed, ask questions and provide feedback to the scrum team on the software presented, which can include criticisms, suggestions for further improvement or a change in direction (Schwaber and Beedle, 2002, p56).

The second part of the meeting (the retrospective) involves the team, without management and the Product Owner (optional) (Schwaber, 2004, p138). This is where the Scrum team and the Scrum Master reflect on the work completed during the sprint to determine what can be improved (Rising and Janoff, 2000; Subramaniam, 2006). Two main questions are asked in this part of the meeting (Schwaber, 2004, p138):

- What went well during the sprint?

- What could be improved in the next sprint?

The team identify areas for improvement, prioritise them and create actionable tasks. They then apply these lessons learned to the next sprint and incorporate actionable into the next planning meeting (Schwaber, 2004, p139).

3.3.5 Tailoring and Implementation of Agile Practices

Each of the previous sections provide guidelines on the implementation and use of three agile practices, which are based on some of the more commonly referenced agile textbooks and literature from early adopters of agile. The adoption of agile practices can bring benefits to ASD teams such as an early return on investment, short time to market, improved quality, enhanced customer relations and better team morale (Barnett and Schwaber, 2004; Williams, Kessler, Cunningham and Jeffries, 2000). However, in reality agile practices are tailored and adapted to suit the needs of individual teams and may not even be consistent within teams in the same organisation as teams adapt the agile practices to suit the needs of their unique environment (Law and Charron, 2005). This is akin to the tailoring and implementation of software development methods, also known as the method-in-action as detailed by Fitzgerald, Russo, and Stolterman (2002) who state that while a formal method is often the basis for implementation, methods are never applied exactly as originally intended. It is reasonable to apply the same interpretation to the implementation of agile practices; namely, that agile practices are rarely implemented based on the textbook definition. The successful implementation of an agile practice depends on the individuals involved, how they interact, work together, communicate, and apply themselves in order to reap the full benefits of agile (Law and Charron, 2005). Consequently, it is difficult to determine what is a ‘good’ or a ‘bad’ implementation of an agile practice as no two ASD teams interpret and apply the same practice in the same way.

3.4 Controlling Agile Software Development Teams

This section first reviews ASD teams, highlighting the difference between agile and non-agile teams and the potential implications from a control perspective. It then examines existing studies on control in ASD teams with the aim of identifying the

type of controls used in such teams and the knowledge gaps in relation to clan control in ASD teams, which this study will address.

3.4.1 Agile Software Development Teams

ASD teams are small, responsive, self-organising, self-managing teams that design, develop, test and deliver software to a customer, who provides continuous feedback (Boehm, 2002; Highsmith and Cockburn, 2001). Yet, it cannot be assumed that by putting a group of individuals together in a team and calling them ‘self-managing’ means they are automatically agile (Moe et al., 2010). While the optimal size of an agile team has been debated, ASD teams are typically small with no more than ten team members (Schwaber and Beedle, 2002). ASD teams are generally co-located, which improves accessibility, communication and collaboration, and reduces the need for formal processes (Cockburn and Highsmith, 2001). Yet, it has been questioned whether agile methodologies are suitable for a distributed team for many reasons as they face additional challenges such as more formal communication, lack of team cohesion, lack of control, possible lack of shared goals, and culture differences (Fitzgerald et al., 2006; Ramesh, Cao and Mohan, 2006). These can be alleviated somewhat by site visits, the facilitation of collaboration and knowledge sharing, and supplementing informal communication with documentation (Ramesh et al., 2006).

ASD teams encourage informal communication and interaction amongst members, have good relationships with the customer, and they practice “leadership-collaboration” instead of a hierarchical command-control management structure (Cockburn and Highsmith, 2001; Highsmith and Cockburn, 2001). Team members should have a range of skills, be cross-functional and have the ability to work together effectively to complete the required tasks (Beck and Andres, 2005; Elssamadisy, 2008). An ASD team is empowered and is responsible for making their own decisions and meeting the goals of each sprint in whatever way it deems appropriate (Schwaber and Beedle, 2002, p45). Therefore, the team set and comply with their own rules, define their own behaviours, and adjust them as required e.g. code must follow coding standards and must be tested before release. However, the team must conform to any existing standards within the organisation such as

coding standards, hardware/software platforms etc. (Schwaber and Beedle, 2002). On occasion, incorrect decisions may be made, which may need to be reversed, but as each sprint is short, decisions can easily be reversed and the impact on the overall delivery of the project is low as it is contained to a small portion of the project (Schwaber and Beedle, 2002, p 46).

ASD teams may be suited to the use of informal rather than formal controls as formal controls are more dominant in a hierarchical structure to allow for the monitoring of performance and behaviour of employees (Ouchi, 1977). ASD teams naturally exhibit self-control as they set their own goals; ascertain the actions by which those goals are achieved; monitor their own work; and reward themselves accordingly (Choudhury and Sabherwal 2003; Henderson and Lee, 1992; Kirsch, 1996). However, it is not obvious if ASD teams exhibit clan control even though team members are dependent on one another to achieve their goals and have autonomy over their own work. It is likely that behaviours values and beliefs emerge from within ASD team, which may have either a positive or negative impact on the team.

3.4.2 Control in Agile Software Development Teams

Software development is a complex task, which involves a wide variety of activities that are interdependent and the use of a variety of controls. As more and more organisations adopt agile methodologies it is imperative that appropriate controls are in place for these teams to deliver successful projects. The increased unpredictability of tasks in such projects can result in more expensive formal controls, which become less effective and less predictable the more uncertain the environment. Management must also develop a better understanding of the factors that help ASD teams to drive project success (Coram and Bohner, 2005). Consequently, the controls that existed for traditional ISD projects may no longer be appropriate, or may require modification for ASD teams as ASD teams work largely on their own. This means that managers must establish some controls to ensure that a project progresses smoothly (Schwaber and Beedle, 2002, p108). These include selecting appropriate personnel for the team, creating an open work environment, encouraging continuous communication with the customer, establishing an evaluation and reward systems

based on team performance, tolerating and anticipating mistakes, and encouraging learning and the sharing and transfer of knowledge (Schwaber and Beedle, 2002, p108).

There are a number of challenges in relation to controlling ASD teams. As mentioned in section 3.2 (p.50) ASD project teams control their own workload, which is a cultural shift for managers who must relinquish control to the team where formal controls, such as formal schedules and project plans are no longer as important (Coram and Bohner, 2005). Management must avoid the imposition of too many formal controls and structure as they may result in unintended consequences (Augustine et al., 2005). At the start of an ASD project teams are formed and individuals with different backgrounds, personalities and skills are assigned to the team who are expected to work together within the context of the organisation to solve problems and deliver solutions (Cockburn and Highsmith, 2001; Beck and Andres, 2005). As agile methodologies rely on collaboration and communication it is important to select the right people for the team to ensure they can work well together people, environment, and organisational culture all influence one another (Coram and Bohner, 2005; Cockburn and Highsmith, 2001). Each project team can develop their own “mini” culture or sub-culture which may be different to that of the organisation (Cockburn, 2007, p135). Cultures may be hierarchical (using a top-down chain of command); random (groups that have no central control); collaborative (groups work by consensus) or synchronous (groups that coordinate without verbal communication – they know what the rules, goals and roles are in specific situations and follow them) (Cockburn, 2007, p136). Each team develops their own values and norms and as the team decides what works for them, this then becomes part of the team culture (Cockburn, 2007, p137; Sliger and Broderick, 2008, p150). However, these values and norms may require adjustment when events occur that affect the team; for example, a change to the working environment (where people sit), introduction of new team member, or the departure of a key team member (Cockburn, 2007, p140). ASD relies heavily on teams and teamwork as opposed to individual work which is prevalent in a plan-driven approach to software development where each individual was solely responsible for their own tasks (Nerur et al., 2005). Therefore, reward systems must be modified to reward team effort as

opposed to individual efforts (Nerur et al., 2005). Consequently, the challenges described by Towry (2003) that relate to the precise definition of required behaviours and measurement of the contribution of each individual to the team are especially prevalent in the context of ASD teams.

3.4.3 Clan Control in Agile Software Development Teams

Agile methodologies are most suitable for projects where the requirements are ill-defined, or where there is a high variability in tasks (Coram and Bohner, 2005). This is an appropriate environment for the implementation of clan control as the goals of the project emerge as the project progresses and task related behaviours are not specified, which is a condition for the implementation of clan control (Figure 2.5.5-1, p.44). As described by Ouchi (1979) in section 2.4.1 (p.20) where behaviours and outcomes are difficult to measure clan control is the preferred mode of control.

It is difficult to apply formal controls such as behaviour and outcome controls to ASD teams as they are less measurable and less observable. As time is required for clan control to emerge naturally and many software development projects work to tight schedules the existence of a clan at the start of an ISD project is unlikely. It may also prove difficult for clan control to develop due to the limited time that project teams work together (Chua et al., 2007). However, it is important to nurture cooperative and collegiate relationships amongst IS personnel, which is a key feature in the development of clan control, to encourage project success (Kirsch, 2004). Chua et al. (2007) support this by finding that clan control is often essential for project success in complex multi-stakeholder ISD projects.

A small number of researchers have studied social capital as a means by which clan control can be quickly implemented in ISD teams in order to increase project success, where social capital is a construct that reflects connections and relationships among individuals (Chua et al., 2007; Kirsch et al., 2010). These studies have focused on ISD teams that utilise the traditional command-and-control structure that exists in most ISD teams, and examine how the project team facilitate clan control, as well as understanding the relationship between the project manager and their team, and how the project manager helps to develop an environment that is

conducive to clan control. Yet, there is little research which addresses how to achieve clan control in ISD teams (Chua et al., 2007) and none to date on how clan control can be achieved in ASD teams.

In contrast to ISD teams ASD teams use a variety of agile practices to help them achieve their goals (Coram and Bohner, 2005). These agile practices allow teams to conduct rituals and ceremonies as described by Ouchi (1980), help to socialise team members into the team, develop shared values and norms, which encourages the team to exert particular behaviours; for example, increased interaction with the customer and with other team members, which are independent of any organisational incentives or rewards (Kirsch et al., 2010). Agile practices are studied to a limited extent with one known study using control theory to understand the conditions under which the use of agile practices help to improve software quality (Maruping et al., 2009). This study builds on this by examining whether agile practices help to achieve clan control in ASD teams and also to make recommendations on how the agile practices can improve clan control in ASD teams.

3.5 Conceptual Framework for this Study

The conceptual framework presented in Figure 2.5.5-1 (p.44) is the foundation for the conceptual framework used in the data collection phase of the study. This framework identified a number of pre-conditions in the literature that must be in place in order to develop clans and implement clan control in organisations. Prior studies related to clan control in large organisational settings whereas this study focuses on clan control in ASD teams where the ASD project team is considered a clan and clan control is regarded as a team phenomenon.

As already discussed in section 1.0 (p.1) the existence of clan control in ISD teams can lead to project success (Chua et al., 2007). The success or failure of such projects, which includes ASD projects has been linked to behavioural factors; for example, how team members work with each other, relate to each other and interact with each other (Robey et al., 1989). It is also suggested that the use of clan control may be more appropriate than formal controls in ISD projects due to the nature of software development, which is often turbulent, complex and non-routine, and the diversity of

the stakeholders and team members involved (Kirsch et al., 2010). Yet, little is known about what facilitates the development of clan control in ISD or ASD teams.

This study argues that a number of the pre-conditions presented in Figure 2.5.5-1 do not have an opportunity to develop within ASD teams for a number of reasons. For example, many organisations work in fast-changing, turbulent environments and do not have the luxury of developing a long and stable membership with their employees (Alvesson and Lindkvist, 1993). Employee turnover in the software industry is not unusual with many IS professionals regularly changing employers (Pee, Tham, Kankanhalli and Tan, 2008; Soon and Slaughter, 2004). This means that in many ISD projects, including ASD projects, team members may not be employed by the organisation for a long period of time, which is a condition for clans to develop (Figure 2.5.5-1 on p.44). Consequently, it is reasonable to assume that several team members were exposed to the cultures and traditions of other organisations and may not accept the cultures and social conditions of their current organisation as quickly as those who are recruited to an organisation for their first term of employment. In addition, due to the nature of ASD projects, participation is often required from a diverse range of stakeholders, who may be internal or external to the organisation (Klein and Jiang, 2001; Pan, 2005). Therefore, it is difficult for ASD teams to evade contact or interaction with other employees or internal business units, or to avoid the influence of culture that exists within other teams within the organisation. All of these restrict the ability of clans to develop naturally within an ASD team.

To help achieve clan control managers or leaders must develop an environment that encourages clan control (Lebas and Weigenstein, 1986), which can include the allocation of appropriate physical resources such as office space, but may also refer to the provision of tools, knowledge and information (Merchant, 1985, p43). However, ASD teams may not always have the necessary resources allocated to their project such as personnel with appropriate expertise, skills, knowledge, and information (Faraj and Sproull, 2000). ASD team members are often assigned to a particular project because they are available and there may be little opportunity for the team to carefully recruit specific individuals for their team that have the desired skills, values and beliefs (Kirsch et al., 2010). Individuals may not have functioned

previously as a team i.e. a group of individuals may be brought together for the sole purpose of developing a particular piece of software (Faraj and Sproull, 2000). Consequently, no pre-defined shared values, norms, or beliefs may exist, although each individual may be familiar with specific values, norms and beliefs that exist within an organisation, which can hamper the development of clan control within a particular team.

It is also argued that some of the pre-conditions for clan control detailed in the conceptual framework (Figure 2.5.5-1 on p.44) may exist in ASD teams, which can be due to the very nature of ASD projects, or through the use of agile practices. Clan control is implemented when there is a lack of prescribed behaviours and outcomes for tasks and where behaviours and outcomes evolve over time as the tasks progress (Ouchi, 1979). This occurs in ASD teams as requirements evolve and change as the project progresses. Also, where individuals are brought together as a team to deliver a software project, behaviours may evolve and change as team members become familiar with one another and each others capabilities. For example, when an agile methodology is implemented behaviours may evolve and change as team members become familiar with the agile methodology and agile practices implemented by the team. These practices such as the sprint planning meeting, the daily scrum and the sprint retrospective meetings can result in the development of certain rituals and ceremonies, to which team members must adhere. The practices also encourage high levels of interaction and communication, which are required for clan control to develop (Barker, 1993; Flamholtz et al., 1985). Yet, there is limited understanding of the social aspect of ASD teams, how individuals within such teams interact with each other (Robinson and Sharp, 2005), how agile practices may contribute to team cohesion (Whitworth and Biddle, 2007a), or how agile practices may help to achieve clan control.

Even though clan control is subtle, illusive and intangible it does requires certain pre-conditions to be present for its successful operation. This study adopts the view that clan control is a people or ‘social’ process and posits that in the context of ASD teams three agile practices play a role in the development of clan control, act as pre-conditions that encourage the development and implementation of clan control, and help to achieve the resultant characteristics of clan control within ASD project teams.

It is proposed that ASD team members can leverage these agile practices to achieve clan control and reinforce various resultant characteristics of clan control such as shared values, norms and beliefs, which may motivate desired behaviours within the team. Therefore, the revised conceptual framework used for this study is presented in Figure 3.5-1, which is then tested by collecting data from three ASD teams (see Chapter 5, p.137 for details on the findings).

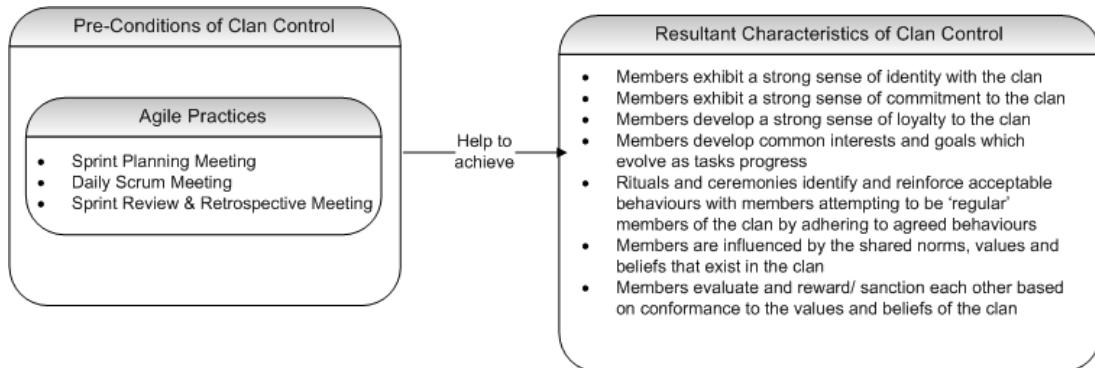


Figure 3.5-1 Revised conceptual framework for this study

3.6 Conclusion

The agile literature, while relatively young, continues to grow with the number of empirical studies increasing each year. However, there are still a large number of gaps in the literature which need to be addressed. Many researchers focus on agile methodologies in general rather than on specific agile methodologies. Two agile methodologies in particular, Scrum and XP, are shown to be the most popular with practitioners, which are described in detail in this chapter. Yet, the number of studies specifically on either of these methodologies is limited, with most using a single case study. This indicates that there is a clear need for further studies on specific methodologies using more than one case study, or using other research methods.

Many organisations tailor an agile methodology or draw on different agile practices from a number of agile methodologies to create a hybrid agile methodology, depending on the requirements of their team or organisation. This demonstrates that all agile practices are not necessarily suitable for each team or project and that there

is a need to develop a better understanding of specific agile practices to establish their applicability to a particular situation. A number of calls exist for further research on specific agile practices to assist with this to which this research will contribute. This chapter focuses on three agile practices: the sprint planning meeting, the daily scrum meeting, and the sprint review and retrospective meeting. Each of these agile practices require the collective participation of all team members, which distinguishes them from other agile practices. They also assist with the management and control of an ASD team and were considered suitable for the study of clan control.

This chapter further examines the literature on control in ASD teams. In ISD teams it can be difficult to rely solely on formal controls due to the diversity of the members and the tasks assigned. This is even more difficult in ASD teams as these teams use agile methodologies to develop software, which facilitate flexibility and continuously changing requirements and tasks. While empirical research exists that uses control theory to provide insights into the management and control of ISD teams and projects, there are few that address the control aspect of ASD with only some notable exceptions. An examination of the literature identified a distinct lack of empirical research on clan control in ASD teams. This mode of control could be very appropriate for such teams as these teams are provided with flexibility to adapt to changing requirements and are empowered to implement the controls they consider appropriate. Research in this area is clearly justified as management must be certain that ASD teams use appropriate controls to ensure the delivery of a successful project.

Clan control in particular focuses on groups or teams and it plays an important part in motivating the behaviours of team members. Therefore, a better understanding of how clan control impacts ASD teams and how they are controlled is imperative as social factors play an important role in such teams. No known research to date has examined how clan control can be achieved in ASD teams. This research uses agile practices as a lens to gain such an understanding and to obtain an insight on the relationship between clan control and agile practices. This chapter concludes with the presentation of a revised conceptual framework that investigates how three agile practices help to achieve the resultant characteristics of clan control in ASD teams

regardless of whether the pre-conditions exist to develop clans or to implement clan control. The findings from the study are presented in Chapter 5.

4 Research Methodology

4.0 Introduction

This chapter begins by re-introducing the research question and objective (section 4.1), which form the basis for the research method and data collection method selected. This is followed by a discussion of the philosophical assumptions underpinning this research (section (c)) and a review of the debate in the literature between quantitative and qualitative studies (section 4.3). It subsequently details the research method selected, the unit of analysis, the data collection methods and analysis utilised and the justification for choosing each (section 4.4 and section 4.5). This section also briefly discusses the benefits of triangulation. The design of the field study is then introduced (section 4.6) which incorporates how the target organisations were selected, the design of the data collection protocol and the details of the pilot study conducted. Section 4.7 details the plan for data collection, the interview process and the process of direct observation. Section 4.8 presents the process of analysing and coding the empirical data and the approach for documenting and writing-up the findings.

4.1 Research Objectives

The research objectives on page 5 of Chapter 1 are re-introduced at this point. Several different control modes were presented in the literature in Chapter 2. This study focuses on one single control mode, namely clan control, which is one of the least studied modes of control. As stated in section 2.5 (p.22) no single definitive framework exists that describes the characteristics or concepts of clan control. The first step in this study is to derive a clan control framework from the literature. The intention is to then use this framework to investigate clan control in the context of ASD teams, which leads to the following research objectives:

- (a) Develop a framework of the pre-conditions and resultant characteristics of clan control
- (b) Use this framework to investigate how agile practices help ASD teams to achieve the resultant characteristics of clan control

- (c) Develop a set of recommendations to identify how agile practices may be improved to achieve higher levels of clan control.

4.2 Epistemological Stance

Research is typically classified based on its philosophical assumptions. The philosophical assumptions are those that relate to the underlying epistemology which guides the research (Myers, 2009, p35). There is a broad spectrum of philosophies for example, positivism, interpretivism, critical, realism, post-positivism (Guba and Lincoln, 1994; Mingers, 2000; 2004; Ngwenyama and Lee, 1997; Orlikowski and Baroudi, 1991; Wolcott, 1990). There has been much debate in the literature on the merits of each. However, two of the most dominant philosophical views in information systems research are positivism and interpretivism with Lee (1991) arguing that the two seemingly opposing philosophical views are in fact “*mutually supportive, rather than mutually exclusive*” and that each is a legitimate approach that researchers should acknowledge. Remenyi et al. (1998) support this by viewing both as related concepts rather than as two extreme, opposing views. Weber (2004) further attempts to clarify both philosophical views and provide an understanding of the fundamental differences between each. He argues that there are a number of commonalities and that the differences between the two philosophies lie mainly in the research methods adopted by positivists and interpretivists. The next two sections detail the fundamental assumptions of each of the two main philosophies followed by the epistemological stance adopted for this study.

4.2.1 Positivism

Positivism originated in the natural sciences (Orlikowski and Baroudi, 1991). It first emerged in the 17th century and evolved from positivism to anti-positivism and later to post-positivism in the 20th century (Hirschheim, 1992). Remenyi et al. (1998, p33) believe positivism first emerged with the work of Auguste Comte that was published between 1830 and 1842, where Comte referred to “social physics”, which suggested that social phenomena could be studied in the same way as scientific or ‘natural’ phenomena. However, regardless of its origins, positivism and its philosophical beliefs has extended to the social sciences and according to Lee (1999) research conducted in the social sciences should follow the same approach as that used in the

natural sciences. Information systems draw heavily on the social sciences and since information systems are fundamentally human and social in nature a positivist approach can be considered an appropriate epistemology for information systems research (Hirschheim, 1992).

Positivists embrace a particular ontological position and typically believe that the researcher is separate and independent of reality and neither affects, nor is affected by the subject of the research (Hirschheim, 1992, p33; Remenyi et al., 1998; Weber, 2004). This approach requires a researcher to remain neutral and detached from the process, so that they do not influence the subject under investigation study. Positivist studies are used primarily to test theory in an attempt to increase the understanding of the event under investigation (Orlikowski and Baroudi, 1991; Walsham, 1995). Large amounts of data are collected that can be analysed statistically to detect trends or abnormalities (Weber, 2004). Propositions or hypotheses are formed, variables are defined, measured and tested and conclusions are drawn from the results, which can be generalised to a specific population (Orlikowski and Baroudi, 1991).

4.2.2 Interpretivism

Interpretivism first emerged through the work of Franz Brentano in the 19th century (Remenyi et al., 1998, p34). This philosophical stance provides different dimensions for research investigation that the positivist paradigm cannot accomplish, which should be welcomed (Chen and Hirschheim, 2004). In contrast to positivism this type of research attempts to understand the context of a particular phenomenon within its natural environment and does not seek to generalise to a specific population (Orlikowski and Baroudi, 1991). It helps researchers to gain a deep insight into how people think and act in various social circumstances and in particular the social circumstances that are personally experiencing (Klein and Myers, 1999; Locke, 2001). Each situation is unique and is specific to the individuals and events involved with interpretive researchers recognising that experiences may differ across different groups or different social settings (Locke, 2001, p9; Remenyi et al., 1998, p34). Interpretive researchers do not predefine variables (Myers, 2009). Instead, they examine a situation in detail in order gain an understanding of what is occurring (Remenyi et al., 1998, p35). With this philosophical approach

interpretivists believe that the researcher and the phenomenon observed cannot be separate from each other; they are naturally intertwined with researchers interpreting what they observe (Weber, 2004).

While the positivist aims for replicability and wants to generalise to the population as a whole, the interpretivist would argue that generalisability is not of crucial importance but that it is important to obtain an in-depth understanding of the phenomenon being examined (Orlikowski and Baroudi, 1991). It is difficult to replicate a study that uses an interpretive philosophy as each study is conducted on a single event at a point in time therefore, generalisation to a population is problematic (Remenyi et al., 1998, p35). However, it is possible to obtain similar findings in additional studies, which can support an emerging theory (Remenyi et al., 1998, p36). Thus, the research methodology used by interpretivists can be equally as valid as long as the data collected by the researcher is available for scrutiny, the context of the research is defined, the research process is clear and the claims made as a result of the research are realistic (Weber, 2004).

4.2.3 Epistemological Stance for this Study

Choosing a philosophical stance at the early stages of research is important, although this is not always a difficult choice for researchers as it may be easily determined by the subject or the research questions under investigation (Chen and Hirschheim, 2004; Remenyi et al., 1998, p37). In business and management positivist research is more dominant than interpretivist research (Myers, 2009, p38; Remenyi et al., 1998, p32) a trend also replicated in information systems research (Walsham, 1995; Chen and Hirschheim, 2004). Walsham (1995) traces the emergence and growth of interpretivist research in mainstream IS journals and conference publications and demonstrates that the number of interpretivist studies in IS research is growing. This is supported by Chen and Hirschheim (2004) who examine the IS literature published between 1991 and 2001. Their review shows that while interpretivist studies, such as case studies have increased, positivist research in the form of surveys still remained the mostly widely used research approach.

For the purposes of this study the research adopts an interpretive philosophical stance. This is partially driven by the research objectives and partially driven by the

nature of the subject under investigation. This study is exploratory in nature and examines ASD teams in their natural setting with a specific focus on clan control. This type of control is intangible and subtle where much knowledge and information within the team is tacit (section 2.5.3, p.26), which would be difficult to identify if a positivist approach was employed. Clan control is also susceptible to the culture of an organisation, its working environment, the influences of internal and external stakeholders, and of members within the clan. Each of these concepts are difficult variable to measure and quantify, which is a necessity for positivist research in order to make propositions and draw inferences about the phenomenon under investigation (Myers, 2009, p39). A positivist study could also find it difficult to establish how these contribute to or facilitate clan control in without having a deep understanding of how the team functions and their relationships with each other.

Positivist research requires a researcher to remain neutral, impartial of the process and detached from the process, so that they do not influence the subject under investigation study (Remenyi et al., 1998, p33). It would inappropriate for a researcher to exclude themselves from this study due to the tacit nature of clan control. This type of control required the researcher to become familiar with different aspects of the team and gain a greater understanding of the team; for example, their working environment, how they behaved, and how they interacted with each other.

In addition, there are few attempts to conceptualise clan control as discussed in section 2.5 (p.22). Some of the concepts presented in Figure 3.5-1 (p.77) are also difficult to conceptualise and do not lend themselves to a positivist study where variables are measurable and constructs are easily defined. Therefore, this research lends itself towards an in-depth study, which can only be addressed by interpretivist research. This type of research presents a perspective that a positivist approach would not be able to achieve (Chen and Hirschheim, 2004).

4.3 Quantitative Vs Qualitative

Research is categorised in many different ways, but the most common classification is quantitative or qualitative research methods (Myers, 2009, p8). Both are equally important and valid and both can be conducted in a rigorous manner, although there

are advantages and disadvantages to each approach (Myers, 2009, p9). Quantitative research methods originated in the natural sciences and are well accepted in the social sciences; for example, survey and experiment (Myers, 2009, p8). Quantitative research is appropriate for large data samples where a researcher wishes to collect data using a pre-determined research instrument that generates statistical data to identify trends and patterns which can then be generalised to a population (Creswell, 2003, p18; Myers, 2009, p9), although in general, many of the social and contextual factors of an organisation or a society; for example, the industry setting of an organisation, are not considered as part of quantitative study (Myers, 2009, p9).

In contrast, qualitative research methods emerged from the social sciences to help researchers study phenomenon in their ‘natural’ context; for example, grounded theory, case study and action research (Myers, 2009, p8). This type of research allows researchers to explore complex relationships and to discover rather than test variables (Corbin and Strauss, 2008, p13). Any concerns with the perceived lack of rigour of such research can be overcome by providing sufficient detail on the design of the study, the method employed, and the analysis and interpretation of the data to convey the rigour and credibility of the study (Lillis, 1999). Qualitative research focuses on a single concept or phenomenon and helps researchers to understand people and the social and cultural contexts within which they function in detail, which can explain why people act in a specific manner (Creswell, 2003, p19; Myers, 2009, p5). Qualitative researchers typically collect data with the aim of developing themes from the data (Creswell, 2003, p18). Qualitative data is useful for answering questions that ask why a relationship is supported, or not, and for understanding the reasons why this may or may not occur (Eisenhardt, 1989). Even though qualitative methods are difficult to replicate (Jick, 1979) it is inappropriate in most cases to generalise to a large population in the same manner that a quantitative method, such as a survey, can generalise to a population (Myers, 2009, p5; Stake, 1994). It is also important to note that qualitative research is not necessarily interpretive; it can be positivist or interpretive, thus the research method chosen is independent of the philosophical position adopted (Myers, 2009, p36).

4.4 Research Approach

A number of different research approaches exist to conduct research in the social sciences, with each having their own strengths and weaknesses (Yin, 2009, p2; Weber, 2004). These include survey, case study, experiment, ethnography, grounded theory and action research (Creswell, 2003, p13; Myers and Avison, 2002, p4). Some of the research methods are predominantly positivist in nature, such as survey and experiment, while others are predominantly interpretive, such as ethnography and action research (Remenyi et al., 1998, p49). The selection of an appropriate method depends on the type of research question asked, the extent of control the researcher has over events and whether the focus is on contemporary or historical events (Yin, 2009, p8). Table 4.4-1 shows how each of these conditions relates to three research methods.

Method	Form of Research Question	Requires Control of Behaviour Events?	Focuses on Contemporary Events?
Experiment	How, why?	Yes	Yes
Survey	Who, what, where, how many, how much?	No	Yes
Case Study	How, why?	No	Yes

Table 4.4-1 Relevant Situations for Different Research Methods (Yin, 2009, p8)

No single research method is more appropriate than any other for all empirical studies (Benbasat, Goldstein and Mead, 1987). The reality is that all research strategies may be appropriate in different situations depending on the research question being addressed. But the limitations of one research strategy can be addressed by another (Kaplan and Duchon, 1988) . For example, ethnography and grounded theory require researchers to spend a long time in the field and require comprehensive observational data from which the researcher attempts to derive a theory (Creswell, 2003, p14). Action research requires a researcher to participate in and study a phenomenon within an organisation, where the researcher intervenes, makes changes, and evaluates the changes, which may require further amendments based on the evaluation conducted (Yin, 2009, p14-18). An experiment, conducted in either a laboratory or in the field, separates a phenomenon from its context and focuses on a number of defined variables, which are controlled and manipulated by

the researcher; and a survey is limited by the number of variables that the researcher wishes to analyse and the context in which respondents can answer the questions asked (Yin, 2009, p14-18). One of the key distinctions between case study and other methods such as experiment and survey is that the researcher does not always know the variables of interest prior to the study or how the variables identified will be measured (Benbasat et al., 1987).

It is also possible for researchers to use a combination of methods in a single study, such as case study and survey, although most studies tend to be more quantitative or qualitative in nature (Creswell, 2003, p4). For example, the main study may use one research method, which is further supplemented by a second research method (Yin, 2009, p63). This mixed-methods approach allows researchers to capture data separately and conduct separate analysis, which is later combined to provide stronger conclusions (Yin, 2009, p63).

When selecting a research approach researchers should select a research method that meets the needs of their study and that is appropriate to answer the question being asked (Moody, 2000; Weber, 2004). Based on the information in Table 4-1 an appropriate research method for the research questions in this study are experiment or case study. However, an experiment typically takes place in a laboratory setting to allow the researcher to control behaviour and precisely manipulate behaviour as required (Yin, 2009, p11). Therefore, it is deemed an inappropriate method for this study.

This intention of this study is to develop a greater understanding of clan control in ASD teams, which requires a deep examination of the views of individuals across a number of teams and how agile practices facilitate clan control. It focused on ‘why’ and ‘how’ and necessitates detailed responses from participants. This topic also requires an insight on how individuals influence each other through informal, clan-like relationships (Adler, 2001), which in this study refers to how agile practices influence individuals to behave in a particular manner, how they influence each other and their relationships with team members. The study is also concerned with an understanding of how context influences the study (Lillis, 1999), which in this study relates to the environment within which an ASD team operates and how their

environment influences clan control. As the focus of this study is on ASD teams working in their natural setting, where the behaviour of individuals is not routine, and the researcher cannot exert control over their behaviour, a case study is considered a suitable approach for this study. Further details on case study approach are provided in the next section.

4.4.1 Case Study Method

Case study research involves the study of organisations in their natural setting to provide a rich understanding of relevant phenomena, thus allowing a researcher to establish *why* something occurs or exists (Merchant and Otley, 2007). It captures and presents empirical data from a real situation describing current challenges or issues that other organisations may face and can identify with, which is one of the main strengths of case study research (Galliers, 1992; Myers, 2009, p80). Yin (1981), who is regarded as one of the most notable authors on case study research, defines a case study as an “*attempt to examine a contemporary phenomenon in its real life context, especially when the boundaries between the phenomenon and context are not clearly evident*”. A phenomenon for a case study may be as broad as a society, or it can be as narrow as a single, collective, social unit such as an organisation, team or individual (Myers, 2009, p74). The case study method is a linear, but iterative process that provides researchers with a structured method for collecting empirical data. It consists of a number of steps: planning, designing the case studies, preparing to collect data, collecting data, analysing data and finally, reporting on the findings of the cases studied (Yin, 2009, p2) where the researcher acts as an investigator (Benbasat et al., 1987).

Case study research is useful when a research subject is broad and complex and requires an in-depth investigation (Dubé and Paré, 2003). It is an appropriate strategy for a researchers who wish to answer a ‘how’ or a ‘why’ question (Benbasat et al., 1987) in an examination of events that are currently taking place rather than historical events, and when behaviours cannot be manipulated or controlled by researchers (Yin, 2009, p11). It helps to capture ‘reality’ in greater detail than is possible by using a research strategy such as a survey or an experiment, because a case is studied in greater depth (Galliers, 1992). It can be used with either the

positivist or interpretive philosophical stance and the approach is not solely limited to qualitative data (Myers, 2009, p77; Dubé and Paré, 2003). It can combine both qualitative and quantitative data from many sources such as interviews, questionnaires and observations (Eisenhardt, 1989; Yin, 1981; Yin, 2009, p19) as was demonstrated by Kaplan and Duchon (1988). It can also develop and analyse more variables than other research methods such as survey and experiment, where the number of variables are pre-determined (Galliers, 1992, p151). Further, it can use a number of different methods to collect data such as ethnography or participant observation (Yin, 1981; Yin, 2009, p16).

Case study research is very versatile and allows researchers great flexibility where they can change direction if a new perspective is revealed (Cavaye, 1996). It has become one of the most common ways of conducting qualitative research (Stake, 1994) and is considered very appropriate where few studies are carried out in an area (Benbasat et al., 1987). According to Myers (2009, p83) each case study should:

- provide some new and interesting information
- display sufficient evidence; for example, supporting quotations from those interviewed
- be complete (i.e. all relevant data has been collected)
- consider alternative perspectives; for example, different theories, cultures, or perspectives
- be written in an engaging manner
- contribute to knowledge

Historically, case study research was often criticised as an insufficient research method due to its perceived lack of rigour. This has been somewhat addressed by researchers in the information systems discipline (Benbasat et al., 1987; Dubé and Paré, 2003; Lee, 1999) and in other disciplines (Eisenhardt, 1989; Yin, 2003), all of whom have contributed to the improvement of case study research as a rigorous approach. Much information systems research published prior to the 1990's in high-ranking journals such as MIS Quarterly and Journal of Management Information Systems, used a single, generally positivist, positivist approach to research. This was highlighted by Orlikowski and Baroudi (1991) who presented a paper arguing that

other philosophical stances, such as interpretive and critical, should be considered in information systems research. This work is supported by Walsham (1995) who discusses and contributes to the emergence of interpretive research in information systems as a valid form of research. While positivist research is not inappropriate it has certain limitations which can be addressed by other forms of research (Orlikowski and Baroudi, 1991). Consequently, case study is now an accepted research method in the IS field (Dubé and Paré, 2003; Klein and Myers, 1999) and is considered well-suited to IS research as IS research generally studies IS within an organisational context (Benbasat et al., 1987; Pare, 2004).

Three different types of case studies are presented in the literature: exploratory, descriptive and explanatory (Yin, 1981). *Exploratory* case studies are typically used in attempt to explain something, to identify factors or problems that may apply in other similar situations, to develop theory or to add to theory (Myers, 2009, p72; Otley and Berry, 1994). They are ideal where research in a particular topic is at an early stage of development (Myers, 2009, p72). The observations sought and the interpretations made from these case studies are dependent on the researcher (Otley and Berry, 1994). *Descriptive* case studies, also called '*illustrative*' case studies are used when the researcher has an initial theory, studies the case and interprets using the selected theory with the aim of reinforcing the theoretical position (Otley and Berry, 1994). There is no theoretical grounding in this type of case study, rather the facts are presented to illustrate a concept (Orlikowski and Baroudi, 1991). The final type of case study is an *explanatory* case study, which has also been called a '*critical*' case study. These type of case studies are used where there is a large body of literature on a topic and researchers attempt to test existing theory, contradict existing theory, compare theories, identify the inadequacies of a theory, or suggest changes to an existing theory (Myers, 2009, p72; Otley and Berry, 1994).

Benbasat et al. (1987) believe that a case study is best suited for developing theories or concepts from the knowledge acquired from participants of the study. Otley and Berry (1994) consider case studies most valuable "*where they are clear about their initial theoretical position and where the researcher consciously attempts to develop their own theoretical modifications*". Therefore, researchers must be clear on their

initial theoretical position, whether they have one, or they develop one as a result of observations made.

The case study method focuses on understanding a single setting in detail and can be used for a number of reasons (a) to provide description, (b) to test theory, or (c) to generate theory (Eisenhardt, 1989). Theory development is an essential part of case study design with many researchers developing a theory (where the literature provides no conceptual framework) or presenting an existing theory from the literature, which is then tested in the case study (Yin, 2009, p35). Eisenhardt (1989) presents a framework that allow researchers to systematically develop theory from case study research. However, due to the volume of data that may be collected in a case, the theory developed may be very complex and attempt to comprise every aspect of the case(s). Further, the theory may be very narrow or idiosyncratic and may not permit the theory to be generalisable at any level. Despite its limitations this approach is valid and applicable in certain situations; for example, when little is known about a phenomenon, or where a fresh perspective is desirable in an existing area of research (Eisenhardt, 1989).

An exploratory case study is used in this research with each case having its own unique characteristics, participants and setting (Stake, 1994). As indicated by Myers (2009, p72) an exploratory case study is ideal where a topic is in the early stages of development. As revealed in the literature review (chapter 2) empirical research that investigated clan control in ASD teams is inadequate. Exploratory studies in this areas are desirable as knowledge in this field of study is scant. This was particularly relevant in this context of this study as each case studied was unique in terms of their location, industry setting, roles present on the team, and experience of team members (Table 4.6.4-1 on p.115). Studying a small number of cases allowed the researcher to examine each team in detail and attempt to understand and analyse the empirical data in their individual contexts. Further, this study attempts to explain how three agile practices help to achieve clan control in ASD teams and identify how these practices can be improved to encourage greater levels of clan control. This required in-depth investigation and interpretation on the part of the researcher in order to understand what happens in reality, which lends itself towards exploratory research.

Many researchers disregard case studies as a valid method for empirical research. One of the main reasons for this is the perceived lack of rigour caused by researchers who do not follow a systematic approach. This was addressed in this study by providing a clear description of the criteria for case selection, information on the research design and on the data collection process as recommended by Dubé and Paré (2003). See sections 4.6 and 4.7 for this detail. Another cause for concern relates to the role of the researcher and the possibility that they may bias the views of participants (Yin, 2009, p14). Section 4.6.5, p.116 details how this issue was dealt with in this study. While these impediments can also exist in other research methods, they occur less frequently than in case research (Yin, 2009, p14). Case research is restricted to a single event and there is difficulty generalising the findings to a population (Galliers, 1992, p151), but if multiple case studies are used it is possible to generalise to a theoretical proposition with each case extending or generalising a theory (Yin, 2009, p15).

Case studies can also take a long time to complete and produce large volumes of data, which must be reviewed and analysed to draw conclusions (Eisenhardt, 1989; Pettigrew, 1990; Yin, 2009, p15). This can be particularly overwhelming for a novice researcher (Myers, 2009, p82). It can also be difficult to acquire the desired level of access in organisations, or where access is initially approved it may be withdrawn at a later stage of the study, which is outside of the control of the researcher (Myers, 2009, p81). In addition, if during the analysis of data conceptualisations arise that are different to those proposed in the conceptual framework or different to the research questions, then this may lead to creation of new research questions, or a necessity to collect new data (Yin, 1981). However, some of these drawbacks can be overcome to some extent by collecting data from multiple sources and triangulating the findings to provide supporting evidence (Yin, 2009, p2). They can also be surmounted by limiting the number of questions or propositions in each case (Yin, 2009, p29).

Multiple data collection methods was used in this study in order to overcome some of these difficulties, but also to gain a greater understanding of clan control (section 4.5 on p.99 for further detail on data collection methods used). As clan control is subtle and intangible (section 2.5.3 on p.26) it can be difficult for individuals to

report (in the context of an interview) on specific behaviours or subtle controls that may exist within a team because they have become the norm within the team and an individual may not consider them relevant or noteworthy to mention. The use of direct observation in addition to interviews allowed the researcher to independently observe the interactions of the team and question particular behaviours when it was deemed appropriate.

4.4.2 Unit of Analysis

An important consideration in a case study is the unit of analysis. For each research problem the unit of analysis must be defined and must relate to the research question(s) asked (Yin, 2009, p29). The unit of analysis refers to what the “case” is. For example, the unit of analysis may be as small as an individual, a specific project, or as large as society (Benbasat et al., 1987; Stake, 1994). If the unit of analysis is a group, data is collected from each individual within the groups, but is aggregated to group level and at the analysis stage data is compared across groups (Sekaran, 2002, p132). In a single case study there may be more than one unit of analysis, if for example, the cases focuses on different groups of employees within an organisation (Yin, 2009, p50). Deciding on the unit of analysis is important as it this can impact on the data collection methods and the sample size required for the study (Sekaran, 2002, p132). When defining the unit of analysis it is necessary to consider whether the findings are generalisable; for example, to other individuals, teams, or organisations (Benbasat et al., 1987). For the purposes of this research the unit of analysis is the ASD team, where the ASD team represent a clan. In the context of this research an ASD team is defined as:

those who participate on a daily basis in the development of a piece of software and use an agile methodology as their software development methodology. It excludes any other individuals or roles who interact intermittently with the team.

4.4.3 Single Case versus Multiple Case Studies

Cases are either single or multiple-case studies with many researchers questioning the contribution of a single case and the possibility for generalisation (Otley and

Berry, 1994). Any number of case studies may be conducted to answer a research question and can include both single- and multiple-case studies (Yin, 2009, p19) with multiple case studies also termed comparative-case studies (Pettigrew, 1990). There is no ideal number of cases that should be studied; it can depend on the quality of the findings, the uniqueness of the findings or their suitability to answer the research questions (Pettigrew, 1990). It may also be related to the availability of resources or to time constraints (Eisenhardt, 1989).

A single-case study is comparable to a single experiment (Remenyi et al., 1998, p181; Yin, 2009, p46). While single case study research has often been discredited this type of study is justified in certain circumstances (Yin, 2009, p46). For example, when a single case is a ‘critical case’ in testing a theory with the intention of confirming, challenging, or extending the theory. It is also applicable where a single case represents an ‘extreme’ or ‘unique’ case where an instance of this case occurs so rarely that it is difficult to establish any commonality with other cases. A third instance is where the case is ‘representative’ of other cases and the findings are designed to be informative. Another situation where a single-case study is suitable is when the case is a ‘revelatory’ case where the researcher gains access to study a phenomenon that was previously inaccessible. The final rationale is if the single-case study is a ‘longitudinal’ case where the study is conducted at different points in time and conclusions are drawn based on findings over a period of time. Before embarking on a single-case study design researchers must ensure that the case is suitable for their needs and will answer the research question(s) and also that it is possible to attain the required level of access (Yin, 2009, p50).

Multiple-case studies are considered a different method by some disciplines. Yin (2009, p53) argues that multiple-case study is a variant of a single-case study and that it follows the same methodological approach. Researchers may consider multiple-case studies comparable to multiple experiments and with each case the researcher attempts to replicate the findings of the previous case, or adjusts the conditions slightly to see if the findings are duplicated (Yin, 2009, p54). Multiple-case studies are used when cases are ‘descriptive’, or when building or testing a theory (Benbasat et al., 1987). They have a number of advantages over single-case design; for example, there are more findings than a single-case study, cross-case

analysis is possible, the study is generally regarded as more robust and the findings are considered more convincing than those of a single-case study, particularly where the findings are replicated across cases (Benbasat et al., 1987; Remenyi et al., 1998, p182; Yin, 2009, p54). However, multiple-case studies can consume a lot of time and resources (Yin, 2009, p53). They are not typically used as a ‘revelatory’ case, a ‘unique’ case or a ‘critical’ case, as these tend to use single-case design. They are used to consolidate or contradict (under expected conditions) the findings of previous cases (Yin, 2009, p54). What is important with multiple-case design is the use of a theoretical framework, which states the conditions under which the phenomenon is found and this framework forms the basis for generalising new cases (Yin, 2009, p54). Where the findings contradict the framework, the results in the modification or extension of the framework.

4.4.4 Research Approach for this Study

Given the lack of prior research on controls in ASD teams and the dominance of studies on specific agile methodologies that used single-case study (section 3.2, p50), a qualitative multiple-case study approach was considered appropriate despite its limitations. It was deemed the most appropriate for this study and the research questions asked for a number of reasons.

Firstly, a number of the concepts investigated were quite subtle and could be considered intrusive to participants. Methods such as a survey would not necessarily capture the level of detail required. As suggested by Myers (2009, p5) the best way to capture detail or to really understand peoples actions or motivations is to speak with people. It was felt that the researcher should establish a relationship with each participant to assist with extracting information on some of the concepts from the participants. It has also been accepted in the literature that it is difficult to conceptualise clan control due to the complexity of the concept (Kirsch et al., 2010). Consequently, developing constructs for clan control is not an easy task, which limited the possibility of using a survey.

Further, as the researcher is examining each case in its own natural environment where participants have control over their own behaviours, rather than in an environment where each variable is controlled, experiments were not deemed a

suitable approach (Benbasat et al., 1987). Other qualitative studies such ethnography and action research require the researcher to immerse themselves in the setting that is under investigation over a long period of time. These types of studies are time consuming and thus, they usually focus on a single organisation (Myers, 2009, p99). Other qualitative studies such as grounded theory develop theory from the data as it emerges rather than collecting data in a structured format that is based on an existing theory or framework (Myers, 2009, 106).

The multiple-case study approach selected facilitated cross-case analysis and permitted an opportunity to examine if the findings were replicated across cases, which provided a foundation for generalisation (Benbasat et al., 1987; Yin, 2009). While each case selected adhered to particular criteria (section 4.6.1 on p.107) that defined the boundary of the case study design the composition of each case varied slightly, which is detailed in Table 4.6.4-1 (p.115). This was intentional to see if the findings were replicated across the cases studies, or if their different circumstances resulted in contrasting findings.

4.4.5 Validity and Reliability

Good research design takes account of validity, reliability and generalisability in order to reduce the risk of getting the answer to the research question wrong (Remenyi et al., 1998, p114). A number of tests are commonly used to determine the quality of research design in the social sciences, which apply to both positivist and interpretivist research (Yin, 2009, p40; Weber, 2004). These are construct validity, internal validity, external validity and reliability (Remenyi et al., 1998, p179). The purpose of these tests is to ensure that the study conducted is rigorous and relevant to the question under scrutiny and that the study can easily be reproduced, which emphasises the reliability of the study (Weber, 2004). These are now discussed as follows.

- *Construct validity.* This type of validity is equally important in interpretivist research as in positivist research. Construct validity ensures that specific concepts are clearly defined for the study and that operational measures for these concepts are identified (Remenyi et al., 1998, p179; Yin, 2009, p40). For example, where data is collected using interviews it is imperative that appropriate

questions are asked and are understandable (Weber, 2004). There are a number of ways in which construct validity can be strengthened; for example, collecting data from multiple sources, establishing a chain of evidence and asking participants in the case study to review draft versions of the case study report (Remenyi et al., 1998, p179; Yin, 2009, p40).

Construct validity was addressed in this study by collecting interview data from each member of the ASD team to obtain different viewpoints, which was further supported by empirical data from direct observations. It was also addressed by carefully developing interview questions from the literature based on each characteristic of the conceptual framework (Figure 3.5-1), which corresponded with each section of the interview protocol (Appendix A). This structure was also followed in the presentation of the findings and analysis (chapter 5). Finally, a summary of the initial findings was communicated to each of the cases for their review and feedback.

- *Internal validity* refers to the “cause-and-effect relationship” between two variables i.e. to what extent independent variable A causes a change in dependent variable B (Sekaran, 2002, p149). This test mainly applies to explanatory or causal studies, rather than to descriptive or exploratory studies, where researchers attempt to explain how and why one event relates to another and may draw conclusions without realising that a third event may have impacted the relationship between the first two events (Yin, 2009, p40). In experiments internal validity is said to be high as variables are investigated in a controlled environment, and the results would indicate the cause-and-effect relationship between the variables under investigation (Sekaran, 2002, p149). A difficulty relating to internal validity is where a researcher draws conclusions, or causal relationships from an event that occurs at a particular time without considering all possible explanations (Sekaran, 2002, p157; Yin, 2009, p40), or ignoring conflicting findings (Eisenhardt, 1989). However, it is possible to overcome these problems during the analysis of the data; for example, by identifying and matching patterns in the data (Yin, 2009, p40) and analysing the data in detail to try to uncover reasons for conflicting data (Eisenhardt, 1989). This may require

the researcher to be very creative or critical in their thinking when analysing the data.

In this study internal validity was addressed by inspecting all the evidence collected and determining if different participants presented converging or diverging views. Where participants communicated similar views, or where interview data was supported by the data gathered from direct observations this strengthened internal validity. Where views of participants conflicted these were also presented as recommended by Eisenhardt (1989), so that confidence in the findings was not reduced. The evidence collected was also analysed for alternative possible explanations as recommended by Remenyi et al. (1998, p180).

- *External validity* relates to the generalisability of the findings of a study to a particular field, although not all findings can be generalised to all other settings or organisations (Sekaran, 2002, p25). For example, field experiments can have high external validity, but have less internal reliability i.e. it is difficult to determine the extent to which variable A alone causes variable B (Sekaran, 2002, p150). Generalisability is of great concern to critics of the case study method, particularly when the research examines a single case as it is not possible to statistically generalise findings from a case study to a population in the same manner as a survey where empirical data has been collected from a sample of the population (Remenyi et al., 1998, p180; Yin, 2009, p38). Each case studied is not a sample (or single respondent) in the same sense as a survey and instead cases should be generalised to a theory (Myers, 2009, p83). Instead, Yin (2009, p38) suggests that '*analytical generalisation*' is appropriate for generating theory from a case study where if the findings from two or more cases support the same theory then replication may be claimed, which is the same logic that underlies the use of experiments. Using a theory in a single case study or studying multiple cases can help to overcome this problem.

Concerns with external validity can only be somewhat addressed in case study research (Remenyi et al., 1998, p180). This study used multiple cases in an attempt to address this problem, but the ability to generalise findings from three

studies is limited. While some findings were replicated across the three cases this only suggests that these findings may also be present in other cases.

- *Reliability.* This test demonstrates that if the study was repeated by another researcher the same findings and conclusions would be reached (Yin, 2009, p40). Reliability can be questioned where the biases of the researcher may be evident, or where there is evidence of inconsistencies in the research process (Weber, 2004). It is much easier to ensure reliability in a positivist study, as the research approaches used are more structured (Weber, 2004). However, some interpretivists argue that each situation is unique and that it is not possible to obtain the exact same set of results at another point in time, therefore, they do not consider reliability an issue (Remenyi et al., 1998, p181). To ensure reliability and avoid bias in case study research it is recommended that the researcher develop and use a research protocol and create a case study database, so that each step is documented, which can be repeated if necessary (Yin, 2009, p40; Dubé and Paré, 2003).

In this research an interview protocol was developed that detailed the objective of the study, provided information on the three agile practices under investigation, and contained the interview questions and corresponding prompts (Appendix A). This interview protocol could be used by another independent researcher to replicate this study. In addition, a case study database was created using qualitative software (section 4.8.1, p.131). This database contained audio files of interviews, interview transcripts, field notes from observations, photographs taken, coded data, and memos as recommended by (Dubé and Paré, 2003). This database could also be used in conjunction with the interview protocol to independently analyse the data gathered.

4.5 Data Collection Methods

Data collection is an integral part of research that can be carried out through the use of several method; for example, questionnaires, interviews, and observation, each of which has its own advantages and disadvantages (Sekaran, 2002, p223). The data collection method(s) appropriate for a study will depend on the research approach selected, the research questions and the unit of analysis (Benbasat et al., 1987). It can

also depend on other factors such as the facilities available, the expertise of the researcher, the time allowed for the study and the costs and resources associated with the collection of data (Sekaran, 2002, p224). Data can be collected directly by the researcher (primary data) or data can be gathered from existing sources such as company records or industry analysis (secondary data) (Sekaran, 2002, p219).

Qualitative research methods such as case studies typically gather data in the form of interviews, direct observations, field notes and documentation to help understand people and their actions (Myers, 2009, p8), but they may also collect quantitative data; for example, through questionnaires (Dubé and Paré, 2003). It is not uncommon for case study research to combine multiple data collection methods in order to capture as much data as is required to answer the research question(s) asked and also to understand the context of the phenomenon under investigation (Benbasat et al., 1987; Yin, 2009, p11). Interviews are one of the most common forms of data collection for qualitative researchers, but documentation such as minutes of meetings, company reports and organisational charts can also be very valuable as they may detail information that individuals may not recall (Myers, 2009). A case study may comprise a small number of interviews with different personnel, but to conduct an in-depth case study it is wise to interview several people to obtain diverse perspectives (Myers, 2009, p79).

Many researchers in the social sciences advocate the use of multiple data collection methods to examine multiple aspects of the same research problem, which provide the researcher(s) with an opportunity to cross-validate data (Jick, 1979) and also create further evidence to support findings and conclusions (Benbasat et al., 1987). This is known as triangulation where a study combines several research methods or more than one data collection method to gather data (Myers, 2009, p10). Triangulation allows researchers to be more confident of their results, which is the main strength of using multiple sources of data collection (Jick, 1979). It allows researchers to examine a topic from different angles (Myers, 2009, p5). It also may help to uncover an unusual phenomenon, or something that deviates from the norm (Jick, 1979). However, it does have shortcomings, as mentioned earlier in relation to replication, but also if the research is not clearly focused the wrong question may be asked, and the outcome may be unsatisfactory (Jick, 1979). Each method should be

equally represented in the data collection, although in some instances one method may be stronger or more appropriate than another (Jick, 1979).

Combining research methods such as a qualitative and a quantitative method is more difficult than combining data collection methods using a single research method (Myers, 2009, p10). Mingers (2001) puts forward a strong argument for using multiple-methods in information systems research. Using multiple research methods and multiple sources of data provides an opportunity for triangulation (Benbasat et al., 1987), which can increase the rigour of the study with the limitations of one method addressed by using a second (Jick, 1979; Kaplan and Duchon, 1988). If the findings from data collected using two different methods correlate strongly then more confidence can be placed in the findings (Benbasat et al., 1987; Sekaran, 2002, p256). Using multiple methods is also a means of eliminating the biases of one research method as those of one research method can cancel out the biases in another research method (Creswell, 2003, p15; Jick, 1979). The remainder of this section will focus on the two data collection methods used for this study, which are techniques well-suited to case study research, particularly exploratory case studies. Interviews were used as the primary source of data collection with direct observation providing supporting evidence. Due to the subtle nature of clan control and the fact that the agile practices were the primary focus of this investigation it was necessary to observe how these practices were used in reality by each of the teams studied (section 4.7.3, p.126) for further detail on how this was conducted). It also allowed the researcher to make the most of their time while present on site with the teams as recommended by Benbasat et al. (1987). Both of these data collection methods are now be examined in more detail.

4.5.1 Interviews

Interviews are frequently used in most qualitative research, regardless of whether the research is positivist, interpretive or critical in its philosophical stance (Myers and Newman, 2007). They are considered the most important data collection method for qualitative research as they provide an opportunity to gather empirical evidence from a potentially wide range of personnel (Myers, 2009, p121; Myers and Newman, 2007). They can garner in-depth information from participants, yet they can also be

staged, where little factual information is acquired (Pettigrew, 1990). With interviews it is important to identify the correct personnel who are capable of answering the questions asked in order to collect rich data (Myers, 2009, p79). According to Myers and Newman (2007) interviews can be placed into one of the following four categories: unstructured, semi-structured, structured and group. But regardless of the type of interview, interviewers should listen, prompt, encourage and direct the interviewee (Myers, 2009, p121). Each of these four types of interviews will now be explained in more detail.

- *Unstructured interviews* do not require the use of pre-formulated questions. The interviewee is free to speak in relation to the topic under discussion, with the interviewer prompting or asking additional questions when required (Myers and Newman, 2007). The objective is to have an informal discussion with participants using broad open-ended questions to help identify some preliminary issues that require further in-depth investigation (Sekaran, 2002, p225). The type of questions asked may vary depending on the interviewee; for example, managers within an organisation may be asked different questions to a worker on the factory floor (Sekaran, 2002, p225). However, it is very difficult to replicate this type of interview across subsequent interviews (Myers and Newman, 2007).
- *Semi-structured interviews* involves the preparation and use of pre-defined questions, which the interviewer follows in the interview although the interviewer may ask additional questions during the interview to acquire clarification on a point or to pursue a new line of enquiry (Myers and Newman, 2007). This type of interview is easier to replicate than an unstructured interview (Myers and Newman, 2007).
- *Structured interviews* are used when the interviewer knows at the outset what information is required (Sekaran, 2002, p227). The interviewer prepares questions in advance and follows the same set of questions, in the same order, across all interviews (Myers and Newman, 2007). This type of interview is most appropriate for telephone interviews, or for conducting market or survey research (Myers and Newman, 2007; Sekaran, 2002, p227).

- *Group interviews*, also called focus groups, permits a researcher to gather views from a number of individuals at the same time on a particular topic (Myers and Newman, 2007). The interviewer may use a structured or unstructured approach, but they are usually restricted to time constraints. Members of the focus group are encouraged to discuss topics with each other and the interviewer facilitates the discussion (Myers and Newman, 2007).

Interviews have their strengths and weaknesses. One of the main strengths of interviews is that the interviewer can gather rich data from each individual on the phenomenon or event under investigation (Benbasat et al., 1987; Sekaran, 2002, p256). It provides the interviewer with the opportunity to adapt questions, clarify responses by asking additional questions, prompt respondents, or rephrase questions when required (Myers, 2009, p125). Interviews should ideally be conducted face-to-face to help establish a rapport with participants. This also allows for the use of visual aids to clarify points (Sekaran, 2002, p251).

However, there are some weaknesses associated with interviews, which must be considered. For example, interviews can take a long time to conduct and costs can increase if interviewees are geographically dispersed (Sekaran, 2002, p251). Researchers may be under time pressure to ask all their necessary questions (Myers and Newman, 2007). Occasionally it can be difficult to elicit responses from some interviewees; some will require a great deal of encouragement to speak whereas others may require little encouragement (Corbin and Strauss, 2008, p28; Sekaran, 2002, p226). Where responses from interviewees are inadequate, rephrasing the question or focusing on a positive aspect of their work may help to relax the interviewee or give them an opportunity to refresh their memory and recall an event (Corbin and Strauss, 2008, p28; Sekaran, 2002, p226). It is also possible that interviewees may feel under pressure to give an answer and they may change their behaviour if they feel intimidated by the interviewer (Myers and Newman, 2007). Or, if an interviewee does not trust the interviewer they may choose intentionally or otherwise to selectively omit information, or to conceal sensitive information during the interview, which may be critical to the study (Myers and Newman, 2007; Miles and Huberman, 1994, p56). Researchers must be aware of these potential difficulties and be prepared to address them. For example, if an interviewee demonstrates

reluctance to be interviewed, the interviewer should consider terminating the interview (Sekaran, 2002, p226). See section 4.7.1 (p.119) for further detail on how the sensitivity of data collection was dealt with and section 4.7.2 (p.123) on how the weaknesses of interviews were addressed.

Interviewer bias can also be a difficulty if the interviewer unintentionally encourages or discourages a certain response through repeated questions or through gestures or facial expressions, or if the interviewer misinterprets a response, which can affect the credibility of the study (Lillis, 1999; Sekaran, 2002, p228). Therefore, it is important for the interviewer to carefully listen to each response and obtain clarification if necessary (Sekaran, 2002, p228). One way to overcome these potential biases is to use multiple sources of evidence, which were used in this study, and to corroborate information from different participants through a process of triangulation (Remenyi et al., 1998, p170) as discussed in section 4.5.

A structured interview was selected for as the primary form of data collection as this method allowed for the collection of rich data, which was necessary to develop an understanding of clan control in the teams studied. It also allowed the researcher to establish a rapport with each participant as each interview was conducted face-to-face with the exception of one interview with a distributed team member. Building a relationship with each participant was important in the context of this study due to the sensitive nature of some of the questions. As the researcher had a number of questions they wished to ask in relation to each characteristic on the conceptual framework (Figure 3.5-1, p.77) it was necessary to pre-define questions. Prompts were also detailed in the interview protocol (Appendix A), which is discussed later in section 4.6.5 (p.116). Some questions asked were closed in nature, but the majority of the interview questions were open-ended in order to provide participants with an opportunity to freely convey their experiences and views on a topic. Further detail on the interview process is available at section 4.7.2 (p.123).

4.5.2 Observation

Observation is a valid form of data collection, although it is often used as a supplement to another data collection method rather than as a primary data collection

method (Cooper and Schindler, 2003, p400). There are three different forms of observation: direct observation, indirect observation and participant observation.

- **Direct observation** occurs when a person is physically present and observes individuals or a situation without any interaction or interference from the researcher (Myers, 2009, p138). In this type of situation a researcher is typically considered an outsider by those being observed (Myers, 2009, p138).
- **Participant observation** requires the researcher to both observe and participate in the event under investigation (Cooper and Schindler, 2003, p406).
- **Indirect observation takes** place when an event is monitored electronically, which can be saved and analysed at a later date (Cooper and Schindler, 2003, p407).

There are a number of advantages to using direct observation as a data collection method. It provides the researcher with the opportunity to personally observe a work environment (Remenyi et al., 1998, p176), which includes listening, reading and taking notes of both verbal and non-verbal behaviour as an event occurs (Cooper and Schindler, 2003, p400). This type of data collection provides researchers with access to groups of people and allows the researcher to identify discrepancies (intentional or otherwise) between what was said in interviews and what individuals actually do on a day-to-day basis (Pettigrew, 1990; Corbin and Strauss, 2008, p29). It allows for the collection of reliable evidence, rather than relying on hearsay evidence from different individuals (Remenyi et al., 1998, p176).

While these advantages exist direction observation requires a substantial time commitment on the part of the researcher(s) (Sekaran, 2002, p252). Also, observation can be intrusive and participants may change their behaviour when they realise they are being observed, although this can be overcome when the participants become familiar with the observer and realise there is no threat (Cooper and Schindler, 2003, p407; Corbin and Strauss, 2008, p29). If a researcher is unsure of the meaning of an event and their resulting interpretation, it is important that the meaning is clarified with participants, particularly when working with different cultures, as the meaning may be mis-interpreted (Corbin and Strauss, 2008, p30). It

is also possible that a researcher unintentionally decides what is or is not important, which can result in notes on one incident or event, but none for another (Miles and Huberman, 1994, p56). In addition, where multiple events occur it can be difficult for an observer to keep track of each event; or where there are multiple observers they may differ on the conclusions they draw from the data collected (Cooper and Schindler, 2003, p406).

For the purpose of this study direct observation was used as a supplement to the primary data collection method of structured interviews. The researcher observed a number of agile practices in person, but did not participate in any of the events. It is possible that team members may not be consciously aware of their behaviours or any subtleties of interactions between themselves and other team members. Therefore, the purpose of the observations was to determine how these agile practices were conducted in reality and also to observe the interactions amongst team members as recommended by Corbin and Strauss (2008, p29). As the focus of this study related to agile practices and how they facilitated clan control it was important to observe these practices in person as these practices required the collective participation of all team members. Further detail is provided in section 4.7.3 (p.126) on how direct observation was conducted in each case.

4.6 Field Study Design

Research design is the plan of how research data is identified, collected and analysed to achieve the research objectives (Yin, 2003, p20). The approach used for this study is illustrated in Figure 4.6-1. This section details how the study was designed, which includes how the cases were selected, a short profile of the cases, the design of the data collection protocol and details on the pilot study that was conducted.

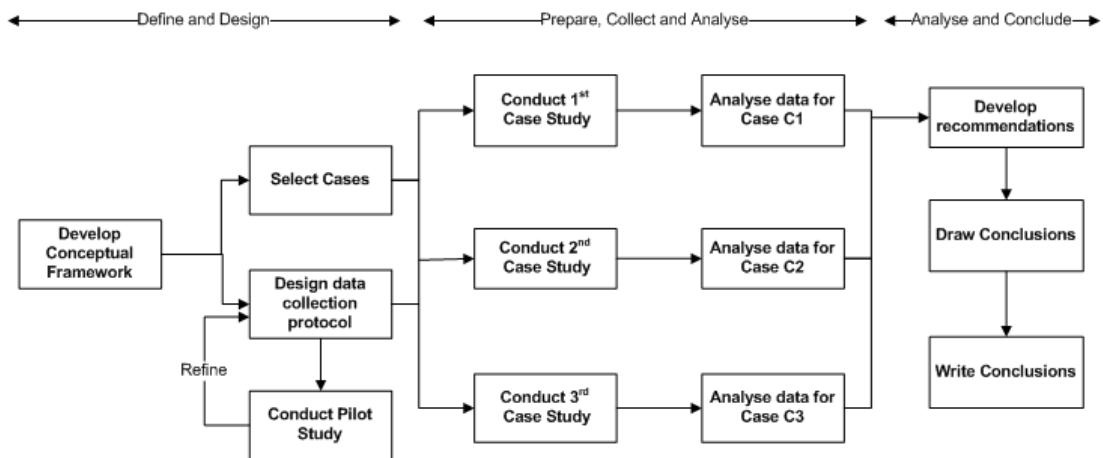


Figure 4.6-1 Case Study Method Adapted from Yin (2009, p57)

4.6.1 Case Selection

A number of factors must be considered when selecting cases, particularly when using a multiple-case study approach (Benbasat et al., 1987). Cases may be selected randomly, although random selection is not preferable (Eisenhardt, 1989). They may be selected because they exhibit extreme circumstances, that are polar opposites, or that have high levels of experience in the subject under investigation (Pettigrew, 1990). Alternatively, cases may be selected to produce similar results (i.e. literal replication), to extend an existing or emerging theory, or to produce contrasting results (i.e. theoretical replication) (Eisenhardt, 1989; Yin, 2009, p55). Primarily, the selection of cases is shaped by the research question (Pettigrew, 1990). If cases are selected that meet specific criteria they define the bounds of the research conducted, help to control environmental variation, and help to generalise the findings (Eisenhardt, 1989). Cases may be selected based on specific criteria; for example, the size of organisations, their industry sector, or whether they operate in the private or public-sector domain. It is also important to ensure that sufficient access can be gained to organisations in order to collect data as this is critical to conduct qualitative research (Yin, 2009, p26). It must be considered that in some instances access to meetings or personnel may be limited (Yin, 2009, p26).

4.6.2 Theoretical Sampling and Saturation

With case study research it is difficult to determine an appropriate sample size. The number of cases may be pre-specified, or they may evolve as the study progresses (Miles and Huberman, 1994, p27). One approach to determining sample size is to conduct theoretical sampling. This allows a researcher to collect data based on different concepts from different participants and settings in an attempt to maximise the opportunity to develop concepts and replicate or identify variations or relationships between concepts (Corbin and Strauss, 2008, p143). The researcher continues to collect and analyse data until a point of ‘stauration’ is reached where each category/theme is explored in different settings and no new data emerges (Corbin and Strauss, 2008, p148). This is the approach that was used in this study as the topic under investigation was new and the number of cases studied provided sufficient depth and breadth in each of the characteristics of clan control.

4.6.3 Criteria used for Selection of Cases

With case study research it is important to set boundaries such as defining the criteria for the selection of cases, detailing specifically what the researcher intends to study or setting a timeframe within which it is feasible to collect data (Miles and Huberman, 1994, p27). For the purposes of this study the following criteria were used to select cases:

- Each case represented an ASD team.
- All team members were available to participate in the study.
- Each of the participating teams had implemented an agile methodology at least six months prior to the collection of data and had continued to use the agile methodology since implementation.
- Each of the participating teams regularly used each of the three agile practices detailed in Table 3.2.2-2 (p.60): the sprint planning meeting, the daily scrum meeting, and the sprint review and retrospective meeting.
- Selection of diverse cases; for example, different geographical locations, different team distribution, different industries, different length of time since implementation of an agile methodology, to determine if the findings are similiar across different settings.

Three cases were selected for this study based on the criteria detailed above. The teams selected were diverse in terms of their location, distribution, and industry setting. They were also selected because they provided in-depth access to a team within their organisation. Full access to all team members was imperative for this study as the team was the unit of analysis (section 4.4.2, p.93) and the team was viewed as a clan. Therefore, it was necessary for all team members to participate in order to obtain all views of the ‘clan’.

Three cases were deemed sufficient following completion of the third case. At that point 25 interviews and 14 observations were conducted (Table 4.7-1, p.119), which yielded rich datasets. Initial analysis of the data resulted in the emergence of similar findings across the three cases. Further analysis of the data consolidated this view and it was reasonable to assume that a point of ‘theoretical saturation’ had been reached where additional insights from collecting further data was expected to be minimal (Eisenhardt, 1989). While additional cases could improve the rigour and validity of the study and potentially increase the breadth and depth of understanding of the phenomenon under investigation it was reasonable to determine that sufficient data was collected for the purposes of this study as the data collected provided considerable insight into the subject investigated (Corbin and Strauss, 2008, p149). Collection of additional data would also result in greater volumes of data, which can be overwhelming for a single researcher to analyse, but it is recognised that the study of additional cases may provide other insights on clan control in ASD teams and this was identified as an avenue for further research in section 7.5 (p.285).

The cases were identified through two different avenues. Firstly, this research was partially funded as part of a wider European project in the area of agile methodologies, which had a number of European industry and academic partners, of which this University was one. One organisation selected to participate in the study was a local industry partner in Ireland based in Galway. This organisation had a number of teams that were distributed and were using an agile methodology for two years. These met the requirement to have a participating team in the study that was distributed and also to have a team where an agile methodology was in use for a long period of time. This organisation was willing to allow the researcher full access to one team in order to collect empirical data.

In an attempt to identify a suitable team from a different geographical location or industry the researcher attended a quarterly meeting of the European industry and academic partners of the project already mentioned in January 2009. Contact was initially established with the second organisation at this meeting, which was an engineering company based in Sweden. Following informal discussions with a manager within the organisation in relation to the research interest was expressed in the research. The researcher informed the manager of the selection criteria for the study and he indicated that such a team was available within the organisation that had recently implemented Scrum. Agreement was made to contact the organisation when the researcher was in a position to collect data.

An attempt was then made to identify a third team, either co-located or distributed. Initial informal contact was made with the Director of Operations of a life assurance company (London-based) at the European Conference of Information Systems (ECIS 2009). This organisation initially expressed interest in the research, but subsequent discussions with the organisation established that none of the ASD teams within the organisation met the criteria defined. However, following discussions internally within the organisation a sister company of this organisation, a software development organisation, based in Dublin, expressed interest in participating in the research. When contacted to discuss the requirements of the research a suitable team was identified within this organisation, which was a multi-cultural, co-located team that were using Scrum for 11 months.

Each organisation and team studied was assured confidentiality prior to participating in the research. There was also an assurance that individual participants or teams would not be named in the research report, or future publications as recommended by Corbin and Strauss (2008, p31). One of the three organisations requested that their identity remain anonymous, therefore, the three organisations will not be named and from this point forward are known as cases C1, C2, and C3.

4.6.4 Case Profiles

The teams were selected for their differences such as their geographic location, the distribution of the team (co-located vs. distributed), the length of time since implementation of an agile methodology, and the cultural diversity of team members

(multi-cultural vs single culture). A summary of the demographic data of the three cases studied is presented in Table 4.6.4-1 (p.115).

Case C1 was an ISD team located in the research and development division of a multi-national financial services organisation. This organisation had implemented a hybrid agile methodology, which incorporated the practices and principles of two agile methodologies (Scrum and XP) in three teams in their Irish offices. The teams were using the hybrid methodology for two years at the time of data collection. Training on the methodology was provided internally within the organisation to the three teams. One of these three teams agreed to participate in this study. The team studied did not use all the practices as defined by the hybrid methodology. Instead, they selected and implemented the agile practices they considered appropriate for their project and were in the process of implementing some additional practices at the time of data collection. However, the team had regularly used the three agile practices required for this study since the project commenced. It is important to note that a number of redundancies had occurred in this organisation in the past year, which was a major event that affected the company and its employees.

At the time of data collection the team was distributed across Ireland, India and the United States. The team consisted of 1 Project Manager (Ireland), 1 Business Analyst (Ireland), 1 Technical Architect (USA), 5 Developers (Ireland) and 2 Quality Assurance (India) personnel (Table 4.6.4-1). There were also a number of customer proxies based in the United States, who represented different customers on the project. Neither the customers nor the customer proxy's were involved in the project on a day-to-day basis. The development team based in Ireland were co-located in an open office space with all team members visible to each other.

The team were working on a multi-year project whose aim was to develop a new IT system that amalgamated five existing IT systems for financial analysts internally within the organisation. The project was very technical and focused on the back-end services for the new software system, which was one of the main reasons the customer/customer proxy was not involved in the project on a day-to-day basis. There was no user interface for the customer to interact with as this was developed by a separate project team based in the United States. At the time of data collection

the project was in existence for two years and was expected to continue for at least two more years. The last release of the software was highly pressurised with the team expected to deliver a large amount of functionality in a tight timeframe. This resulted in a significant amount of overtime for team members in order to achieve the deadline. In contrast, the functionality required for the current release is not as clearly defined, is quite fragmented and incorporates a substantial portion of design work with the team having ample time and resources available to complete the tasks for this release.

Case C2 was a multi-national engineering company focusing on power and automation technologies for utility and industry customers. The team studied used the Scrum methodology to develop software and were the only team within their division that used an agile methodology to develop software. The decision to implement Scrum was driven by the manager of the team prior to the commencement of the current project. This manager joined the team from another part of the organisation two years prior to the data collection and over that time has slowly changed the way the team work, which included the implementation of Scrum. Training was organised for team members on how to implement and use Scrum and one team member was certified as a Scrum Master. The team implemented and adopted Scrum and have regularly used the three agile practices required for this study since the project commenced.

The team studied was co-located and based in one of the Swedish offices of the organisation. At the time of data collection the team consisted of 1 Scrum Master, 1 Product Owner, and 7 Developers (Table 4.6.4-1, p.115). All team members were Swedish and had previous experience of developing software in a non-agile environment. The customer for the team was an internal department within the organisation who was represented in the team by the Product Owner who was involved in the project on a daily basis.

The team worked on three different projects simultaneously, which was challenging for the team. The main project related to the development of one component of a generic software platform for a software product, which was delivered to an integration team rather than to an end-user. The team were also expected to complete

tasks in relation to the maintenance and support of two other software systems when required and deliver these to multiple stakeholders. These tasks related to features or defects in the software that required correction. These tasks were external to the work assigned to individuals during each sprint and were generally not included in the sprint planning, but on occasion they were included in the sprint planning with team members expected to work on three different projects in the one sprint. This was confusing for the team in addition to the challenges associated with switching from one project to another in a short timeframe, which could affect the productivity of the team. Where possible the team tried to assign tasks for each project to particular individuals to avoid switching projects mid-sprint. The team were currently trying to minimise this through the Product Owner who was under pressure from a number of different personnel in terms of prioritising projects and tasks for the team to deliver. At the time of data collection the project was in existence for 9 months and was expected to continue for an indefinite period of time.

Case C3 was a team based in an organisation that developed software products for the life assurance industry. This organisation, initially established in Ireland, was in operation for approximately 20 years prior to its purchase by a multi-national insurance company in 2007. Currently, the office in Ireland is one of a number of regional offices located worldwide. While the team studied was part of a large multi-national organisation the office in Ireland remained small with approximately 50 employees employed. The team studied was the only team in this division that used Scrum to develop software.

The implementation of Scrum was driven by the Director of Product Development who was recruited by the organisation 14 months prior to data collection. His experience in other organisations had involved agile methodologies and he was aware of the benefits of using agile. He slowly began to introduce agile practices and thinking to his team, which was consolidated by the provision of training on Scrum for the entire team. Following the training various agile practices were implemented, which included the three agile practices required for this study. Two team members were certified as Scrum Masters, which allowed for the division of the software team into two teams when required. At the point of data collection Scrum was in use for 11 months and both Scrum Masters were part of the same team as the tasks for the

next phase of the project were merged. One individual acted as the Scrum Master for this phase of the project, whereas the other held a developer role. It was expected that the teams would separate again in future phases of the project and two Scrum Masters would be necessary.

The team studied consisted of 1 Scrum Master, 1 Product Owner, 5 Developers and 1 Quality Assurance personnel (Table 4.6.4-1). The team was multi-cultural, consisting of a number of nationalities (English, French, Hungarian, Irish, Italian and Spanish). All team members had prior experience of developing software in a non-agile environment. The team was co-located in an open office space with all team members visible to each other. The organisation had external customers (insurance companies and financial institutions), but from the team's perspective their main customer was an internal team of underwriters who communicated with and represented the needs of external customers. On a day-to-day basis the underwriting team was represented by the Product Owner who actively participated in the agile team.

A summary of the profile of the three cases is presented in Table 4.6.4-1.

	Case C1	Case C2	Case C3
Organisation Location	Galway, Ireland Boston, USA Bangalore, India	Västerås, Sweden	Dublin, Ireland
Industry Sector	Financial Services & Investments	Engineering	Software Development
Multi-National Organisation	Yes	Yes	Yes
Team Distribution	Distributed	Co-located	Co-located
Team Culture	Multi-cultural	Single culture	Multi-cultural
Team Size	10 ¹	9	8
Team Composition	1 Project Manager 1 Business Analyst 1 Technical Architect 5 Developers 2 Quality Assurance	1 Scrum Master 1 Product Owner 7 Developers	1 Scrum Master 1 Product Owner 5 Developers 1 Quality Assurance
Average years software development experience	11 years	14 years	10 years ²
Average years employed by the organization	4 years	15 years ³	5 years
Agile Methodology Implemented	Hybrid of XP and Scrum	Scrum	Scrum
Sprint Duration	3 weeks	2 weeks	2 weeks
Length of time since agile implementation	2 years	9 months	11 months
Customer	Internal, but distributed from the core team	Internal	Internal Representative External customer – insurance companies

Table 4.6.4-1 Profile of participating organisations and teams

¹ The two quality assurance team members based in India were part of the team when data collection commenced. They subsequently departed from the team and were not replaced prior to the completion of data collection.

² One individual had 30 years experience in the software industry. The remaining team members had between 3 years and 11 years experience in the software industry.

³ One individual has been employed by Case C2 for 30 years, but worked as an electronic engineer for the first 15 years. This is included in the calculation.

4.6.5 Design of Data Collection Protocol

The primary data collection method used was structured interviews. Consequently, an interview protocol was developed (Appendix A), which was utilised in each interview. This consisted of a list of pre-defined questions devised from the literature, based on the characteristics presented in the conceptual framework in Figure 3.5-1 (p.77). When creating the interview questions the language of the questions was considered to ensure there was no ambiguity, as recommended by Myers and Newman (2007). Questions were clearly phrased and the language and terminology used was simple and non-academic. Long and complex questions were avoided as these can easily be misunderstood (Cooper and Schindler, 2003). Also the researcher was careful to ensure the questions were not leading, or did not introduce bias on the part of the researcher. This was achieved by ensuring that the interview protocol was designed to ensure full coverage of each element of the conceptual framework presented in Figure 3.5-1 with each question designed to be neutral and included pre-specified prompts where required as recommended by Lillis (1999).

The interview protocol was divided into a number of sections. The first section detailed the objective of the study and listed the three agile practices and a description of each practice as detailed in Table 3.2.2-2 (p.60). A copy of this section of the interview protocol was provided to each interviewee for reference during the interview. The remainder of the interview protocol was solely for the use of the researcher. The second section used closed-questions to capture background details on the interviewee such as how many years experience each participant has in the software industry. This was followed by a small number of mostly closed-questions questions in relation to the team and the project. The remaining sections of the interview protocol asked a number of open-ended questions in relation to each of the characteristics presented in Figure 3.5-1 (p.77) and how the three agile practices facilitated each of these characteristics. Each section contained a number of potential probes that could be used to further explore the characteristics. Each section followed logically from one to the next. It was not anticipated that any topic would be omitted by the interviewer, but it was possible that the order of the topics could change depending on the flow of the conversation. The questions were designed with this in mind and allowed the interviewer to return to a topic that was skipped.

The questions devised were predominantly open-ended in nature. These type of questions are better for eliciting sensitive information as they encourage discussion and allow participants to freely respond to the questions. A small number of questions were closed, whose objective was to capture specific data; for example, how long an individual was employed by their organisation. Prompts were also created for each of the characteristics of the conceptual framework, solely for the use of the interviewer, and these were documented in the interview protocol (Appendix A). The purpose of the prompts was to assist the researcher if an interviewee had little to say, or did not know how to answer the question. They were also used to elicit additional responses from interviewees on a particular characteristic.

4.6.6 Pilot Study

Following the completion of the interview protocol a pilot study was conducted with an individual from industry as recommended by Benbasat et al. (1987) when the research conducted is exploratory. The purpose of the pilot study was to confirm if the unit of analysis is appropriate and also to familiarise the researcher with the subject under investigation (Benbasat et al., 1987). The pilot study also allowed the researcher to test the questions for ambiguity and to time the duration of the interview. It was desirable on the part of the researcher for the interview to last no longer than one hour as it was felt that this was a sufficient commitment on the part of each interviewee.

This individual approached to participate in the pilot study had 20 years experience in the software industry and had worked in an organisation where agile methodologies were in use for a number of years. This person had experience of using agile methodologies in ISD teams where the teams used a hybrid of XP and Scrum. The individual interviewed was familiar with and had previously participated regularly in an industry setting in each of the three practices studied. This person willingly agreed to participate in the pilot study. A date and time was agreed on which the pilot interview would take place.

At the start of the pilot interview permission was requested to record the interview, to allow the researcher to listen to the recording at a later stage and identify areas for improvement; for example, where the researcher did not clearly explain a concept, or

where the order of the questions were not appropriate, or where two questions were asking for similar answers, but in a different way. Permission was granted by the interviewee to record the interview. A one-page document detailing each of the agile practices as shown in Table 3.2.2-2 (p.60) was provided to the interviewee for reference during the interview. The interviewee was asked to indicate which of the practices were used by the team they had worked with and to consider these practices when answering the questions posed.

Prior to the pilot study it was the intention of the researcher to study all the agile practices (Table 3.2.1-2 and Table 3.2.2-2) determined by which agile methodology was used by the ASD team studied. However, as the interview progressed it became obvious very quickly that the interviewee was struggling to consider all agile practices when answering each question. This was partially because the team did not use all the agile practices as defined by the methodology and also because the individual had never used some of the agile practices, or had used them very infrequently. The tendency of the interviewee was to continuously refer to the sprint planning meeting, the daily scrum and the sprint review and retrospective when answering questions. When asked why these three practices were constantly referred to the respondent indicated that these were the practices that the interviewee was most familiar with and felt comfortable discussing.

Consequently, following the completion of the pilot interview the researcher reconsidered the number of agile practices that should be studied. A decision was made to reduce the number of practices to three as it was felt that this was a suitable number for an interviewee to focus on at any one time when answering each question and also these three practices required the participation of all team members. In addition, after listening to the recording and reviewing the transcript of the interview some of the interview questions were revised to ensure clarity of the questions and the order in which other questions were asked was modified, so that they followed a more logical train of thought. The duration of the pilot interview was 87 minutes, which was longer than anticipated. Therefore, some questions were removed or were replaced as a prompt in another question as the number of questions were excessive for the timeframe anticipated for each interview. This revised version of the interview protocol was then used for the main data collection (Appendix A).

4.7 Data Collection

This section presented the data collection process. It first details the planning of data collection, which includes considerations in relation to the sensitive nature of the data and the existence of tacit knowledge amongst team members. This is then followed by details on the interview process and how direct observations were conducted. Table 4.7-1 details the data collected in each case.

	Case C1	Case C2	Case C3
Number of Interviews	8	9	8
Roles Interviewed	1 Project Manager 1 Business Analyst 1 Technical Architect 5 Developers	1 Scrum Master 1 Product Owner 7 Developers	1 Scrum Master 1 Product Owner 5 Developers 1 Quality Assurance
Number of observations	5	4	5
Observations	2 Sprint planning meetings 1 Daily scrum meeting 2 Sprint review and retrospective meetings	1 Sprint planning meeting 3 Daily scrum meetings	1 Sprint planning meeting 2 Daily scrum meetings 2 Sprint review and retrospective meetings

Table 4.7-1 Data collection in the three cases studied

4.7.1 Planning Data Collection

The collection of data in each organisation was planned and agreed in advance with a manager within the organisation. Empirical data was collected primarily by face-to-face interviews, which was supplemented by direct observations of agile practices. Data collection took place over a seven months period from December 2009 to June 2010. Even though C1 was close in proximity to the University the data collection was sporadic and took place over a period of 4 months. This was partially due to availability of personnel and partially due to a request by management to minimise disruption to the team. In C2 and C3 arrangements were made with the relevant manager for the researcher to be present in their organisation for four consecutive days to collect data. This was to minimise travel expenses to these organisations, but also to develop a relationship with the teams and become familiar with their working

environment, behaviours and interactions with one another. Data collection in C1 was completed before it commenced in C2 and likewise data collection in C2 was completed prior to the commencement of data collection in C3.

In each case the coordination of the interview schedule and the observations was assigned to one individual. The researcher communicated their requirements to this person who drafted a timetable for the period of the visit, which allowed appropriate time to conduct interviews and to observe the agile practices. The agile practices were scheduled for their normal day and time. Interview dates and times were agreed with team members to take account of their availability and to minimise disruption to their work. There was a natural hierarchy in each team from Project Manager/Scrum Master/ Product Owner through to Developers and QA. Developers and QA were selected for interview first, which were conducted in no particular order, and the interviews concluded with the Project Manager or Scrum Master and Product Owner. This helped to identify patterns within the development team, which were then consolidated with the views of what were considered higher ranking roles. On occasion the views of the more senior roles on the team differed from those of the team. Where this occurred this is reported in the findings.

A meeting room was reserved by the coordinator for the dates and times allocated, so that the interviews could take place in private. All but one of the interviews were conducted face-to-face in a private meeting room at the participants place of work. The remaining interview was conducted with a distributed team member (C1), who was located in the United States. This person was interviewed via a conference call, which was organised by the coordinator and was conducted in the offices of this organisation who facilitated the conference call.

Sensitivity of Data Collection

Collecting sensitive data from respondents can be problematic especially if respondents are reluctant to reveal information, or if they provide an inaccurate response (Dalton, Daily and Wimbush, 1997). Due to the sensitive nature of some of the data collected anonymity and confidentiality was provided to each participant in an attempt to avoid this as recommended by Corbin and Strauss (2008, p31). Where a participant demonstrated reluctance to answer a question, or was obviously

finding it difficult to answer a question without divulging certain information the researcher allowed time for the interviewee to gather their thoughts and words so they could express themselves. They were also reminded that what they said in the interview was confidential to help put them at ease. The validity and reliability of the data collected was also verified to overcome biases of the researcher. Where more than one person reported the same or similar experiences, or where the researcher observed an event, that corroborated what was reported by one or more respondents (Yin, 2009, p42) this was identified and it helped to strengthen the findings and validity of the study.

Researchers should ideally be objective when collecting qualitative data, although it is recognised that this is difficult as each individual will have their own perspective, knowledge and experience upon which they will naturally draw when collecting data (Guba and Lincoln, 1998; Miles and Huberman, 1994, p8). Where a researcher has professional experience in the field this can enable them to understand something more quickly as they may be familiar with the situation or event, but it may also affect their interpretation of the data (Corbin and Strauss, 2008, p33). Despite this their experience and knowledge may also allow a researcher to identify connections between concepts that they may not have otherwise recognised (Corbin and Strauss, 2008, p34). This researcher has 12 years professional experience in the software industry, which did assist in an understanding certain situations or feelings of the team; for example, difficulty working on tasks for multiple projects at the same time. However, the professional experience of the researcher related to non-agile teams. Therefore, the researcher did not have any preconceptions or experiences of using the agile practices studied. Instead, the focus in this study was on what the participants said or did as a result of using agile practices.

As the number of interviews conducted and observations of agile practices increased the researcher did become more sensitive to what was being said and observed. This was particularly prevalent in C2 and C3 where the researcher was immersed within the team for a number of consecutive days. The continued presence of the researcher in the organisation for a number of days helped to develop relationships with team members; for example, through informal conversations held at coffee breaks or at lunch time. Specifically, these informal conversations with different team members

assisted the researcher in developing an understanding of the Swedish culture in C2, which was different to that of the researcher. As noted by Corbin and Struass (2008, p35) where research is cross-cultural sensitivity to culture is important and the researcher must attempt to understand the culture within which the research is conducted.

Tacit Knowledge

Identifying tacit knowledge is difficult as it is undocumented information that is known to individuals who interact with each other (Polanyi, 1962). Observation, face-to-face interaction, or informal social networks are means by which tacit knowledge is transferred between team members (Wyatt, 2001; Gopal and Gosain, 2010). Another means is that of socialisation into a team or clan through the use of clan control (Turner and Makhija, 2006). To fully understand the existence of tacit knowledge within a team it requires a researcher to immerse themselves repeatedly in a team and question what they do (Baumard, 1999, p4). Over time it is then possible for a researcher to identify tacit knowledge and document it (Wyatt, 2001) with team meetings used to share and acquire knowledge and information (Walz et al., 1993).

In an attempt to understand tacit knowledge within the teams studied the researcher immersed themselves within the teams for a number of days, particularly in cases C2 and C3, and directly observed behaviours and interactions amongst team members. Non-participative direct observations were made of the agile practices by the researcher, in an attempt to avoid influencing the interactions of team members. But, it is possible that the presence of the researcher impacted on how the teams functioned, which is recognised as a limitation of this research (section 7.4.2, p283). Some team members were questioned informally following the observations to clarify why certain actions occurred to try and extract tacit knowledge that could be documented, which could provide insights into how the teams operated. However, it was difficult to accurately determine if the information captured was actually tacit knowledge, or if it was more commonly known outside of the team environment.

4.7.2 The Interview Process

To establish credibility and to help build a rapport with the interviewees all but one of the interviews were held face-to-face. The remaining interview was conducted via conference call with a distributed team member based in the United States. The researcher introduced themselves at the start of each interview and provided a brief background on herself, the research, the objective of the study, what the researcher hope to achieve on completion of the study, and how the findings would be presented. Each interviewee was informed that the interview would last no more than one hour. Permission was requested to digitally record the interview as recommended by (Sekaran, 2002, p231) which was willingly granted by all participants. This minimised note-taking and allowed the researcher to concentrate on asking questions and listening to the responses. Anonymity was guaranteed to each participant who was also informed that their responses to questions were confidential and the digital recordings were solely for the use of the researcher to assist with analysing the data collected. They were also assured that no single participant would be personally identified in the study. This was essential for this study due to the sensitive nature of the responses required from participants for some questions.

Each participant was provided with a paper copy of the three agile practices as detailed in Table 3.2.2-2 (p.60), which also contained an explanation of each agile practice. The participant was able to refer to this during the interview and it also reminded them to consider each agile practice when answering interview questions. The distributed team member based in the United States was emailed a copy of the three agile practices as shown in Appendix A along with a brief explanation of the purpose of the research in advance of the interview. The interviewee was requested to bring this material to the interview for reference to which the interviewee complied.

The researcher proceeded with the interview and asked the participants each question in turn as detailed in the interview protocol. Each interviewee was given an opportunity to respond to questions frankly. As the interviewee responded the researcher sometimes probed further and asked additional questions using the

prompts detailed in the interview protocol. Interviewees were also encouraged to provide examples from their own experiences to clarify a point made in order to gain a better understanding of what the participant was discussing. The prompts were used if a participant did not expand sufficiently on a point, or where the response of the participant was brief or inadequate. Where the interviewee digressed for a period of time from the topic under discussion the researcher was careful to allow some time for the interviewee to make their point, but then directed the conversation back to the interview questions. All prompts were not used in each interview as interviewees were allowed to speak freely and several interviewees responded at length on several questions. This reduced the amount of time available to ask questions relating to all prompts as the researcher had indicated that the interview would have a duration of no more than one hour.

Each interviewee was asked to consider three agile practices when answering a question and occasionally interviewees tended to focus on one practice more than the others. Where this occurred the researcher was careful to draw attention to the remaining practices before proceeding to the next question.

As some of the interviewees were of different nationalities they sometimes had difficulty understanding a question asked even though all interviewees had a very good understanding of English and spoke English well. When this occurred the researcher took the time to rephrase the question, ask the question in a simpler format, or provide further detail to explain the concept so that the interviewee could understand. On the small number of occasions where the response of an interviewee appeared to be reluctant, or where interviewee was struggling to answer a question without divulging all the relevant facts the researcher gave the interviewee ample time to answer the question. Where required the interviewer rephrased the question in a slightly different way, so that it did not cause as much difficulty for the interviewee to respond.

Each interview lasted approximately one hour, with interview times ranging between 50 and 80 minutes. Where the interview exceeded one hour the researcher enquired at the end of the hour if the interviewee was willing to continue with the interview in order to answer all questions. All interviewees agreed to continue with the interview.

At the end of each interview the researcher thanked the interviewee for participating and informed them that the collective response of the team would be communicated to their manager at a point in the future. The digital recorder was then switched off. Prior to departure from the interview each participant was requested not to divulge the contents of the interview to other team members. This was an attempt to reduce the possibility of other team members discussing and pre-preparing answers to questions asked. Each participant agreed to comply with this request.

Once the interviewee left the room the researcher made notes of initial impressions of participants and their responses as recommended by Eisenhardt (1989) and as they may be useful when reviewing the data collected. This was imperative when collecting data from multiple persons as data may be lost or forgotten over time if important observations are not noted immediately following data collection (Benbasat et al., 1987). This was particularly important when more than one interview was conducted in a single day, as according to Sekaran (2002, p231) when the number of interviews increases it can be difficult to recall who said what. It is also recommended to have a period of reflection after each contact to allow the researcher to consider the main concepts and themes discussed and to determine what was learnt (Miles and Huberman, 1994, p51). In this study the period of reflection between most interviews was very short (i.e. one or two hours) due to the time constraints for gathering data as discussed previously. However, the notes made by researcher following each interview provided an opportunity to adjust the interview protocol slightly when required; for example, to include an additional prompt for a particular question, based on a piece of information provided by an interviewee. In a small number of interviews additional conversation occurred once the digital recorder was turned off, which in some instances related to what was already discussed and in other cases were some afterthoughts that the interviewee wished to share. Where this occurred notes were taken immediately following the conclusion of the conversation as recommended by Corbin and Strauss (2008, p28).

In C2 and C3, the researcher provided initial feedback at the end of the visit. This consisted of feedback in the form of a short presentation, which summarised a number of items, both positive and negative, that were raised by team members. In C2, this presentation was made to the Product Owner and the Scrum Master. In C3,

the presentation was made to the Development Manager and the Director of Product Development. A copy of the presentation was later emailed to each of these individuals for information purposes. In the three organisations the researcher informed their main contact that copies of the analysis of findings and also copies of any publications would be made available for review.

All prompts were not used in each interview. This occurred where an interviewee responded at length to several questions/prompts and the time limitations for each interview required the researcher to progress with the remaining interview questions to ensure that all questions were asked to each interviewee.

The interviews recorded in C1 were transcribed following each interview. As the interviews in C2 and C3 were conducted over a very short period of time these were transcribed after all interviews were completed. The interviews were transcribed by a number of individuals - the researcher, a research assistant and an organisation that offered a transcribing service. The total time for transcription was approximately five weeks. These were subsequently coded using software for qualitative research (NVivo), details of which is provided in section 4.8 (p.131).

4.7.3 Direct Observation

Data was also collected in each organisation by observing a small number of agile practices in action. This was used as a supplement to the primary data collection method, but it was equally important. The number of observations was limited due to amount of time that the researcher was present in each organisation. Sprint planning meetings occurred at the start of each sprint, which were held every two weeks, therefore it was only possible to attend one of these meetings in C2 and C3. Even though these observations could be considered intrusive and threatening for a team (Cooper and Schindler, 2003, p407; Corbin and Strauss, 2008, p29), the teams were open to being observed. In C1 and C3 a small number of team members were interviewed prior to the first observation and were familiar with the researcher. As the number of observations increased this also helped to build relationships between the researcher and the team and the team did not appear to feel threatened. Most team members were also quite experienced (Table 4.6.4-1, p.115) and were quite

confident in themselves and their abilities. The experiences of the researcher may have differed if the team observed had little industry experience.

The observations of sprint planning meetings occurred in C1 at the start of two different sprints. In each case scrum meetings took place daily, with the exception of the first day of the sprint when the sprint planning meeting took place. The sprint review and retrospective meeting occurred at the end of each sprint. This agile practice was not observed in C2 as this meeting occurred on the Friday prior to the arrival of the researcher. The sprint review and retrospective was observed in both C2 and C3. Following each observation the researcher clarified with participants the meaning of certain events where required as recommended by Corbin and Strauss (2008, p30). For example, at the daily scrum meetings in C2 it was observed that some team members revised the amount of time required to complete a task on the Scrum board, but others did not. When questioned about this after the meeting it transpired that if the estimate for a task decreased or remained the same the estimate was updated, but if the time estimate increased, no change was made. This had become the norm within the team.

In C1 agreement to directly observe these practices was reached in advance with the project manager, who facilitated the researcher and communicated a number of suitable dates and times to the researcher on which they could attend the practices. The researcher then selected times that were most suitable for the data collection process. As the sprint planning meeting and the sprint review and retrospective occurred once every three weeks, the researcher selected dates during the period which the interviews were conducted. This was to assist with the validation of interview findings against the observations made. The agile practices were conducted amongst team members in a private meeting room. Prior to the commencement of each agile practice, the researcher was introduced to the team by the project manager and the team were informed that the researcher was observing the practice for research purposes. The researcher sat at the side of the room to observe the agile practice in action, so as not to interfere with the usual format of the meeting. Direct observations of the interactions amongst team members were noted and documented. Documented observations included notes on the date, time and location of the observation, the physical setting, where team members sat/stood,

tools used by the team, behaviour at meetings, and levels of communication and interaction between individual team members. The observations lasted between 30 mins to two hours for sprint planning meetings; between 10 and 15 minutes for daily scrum meetings; and between 30 minutes to one hour for sprint review and retrospectives.

In C2 the researcher was present at a number of agile practices for observation purposes and took notes during each observation. The researcher was first introduced to the team at the sprint planning meeting, which was held on a Monday morning. The Scrum Master explained briefly to the team that the researcher would be with the team for a number of days, observing various agile practices and interviewing all team members. Each of the agile practices observed were conducted in Swedish, the native language of team members, although all documentation such as tasks detailed in Excel were detailed in English. As the team communicated in Swedish body language and the tone of voice were an important indicator when team members disagreed with another, or where all team members were in agreement with each other. Details of the work environment and of the agile practices were also noted by the researcher; for example, where people sat/stood in relation to each other during these meetings, how the meeting area was laid out, the tools available to the team. Permission was also granted to take a small number of photos of the daily scrum board. An example of the Scrum Board and the burndown chart used by this team are displayed in Figure 4.7.3-1 and Figure 4.7.3-2.

Informal conversations were also held with different individuals during a lunch break or a coffee break and any observations made or comments made by team members were documented by the researcher. Each of the documents containing field notes from observations were subsequently coded using software for qualitative research (NVivo), details of which are provided in section 4.8.2 on p.131.

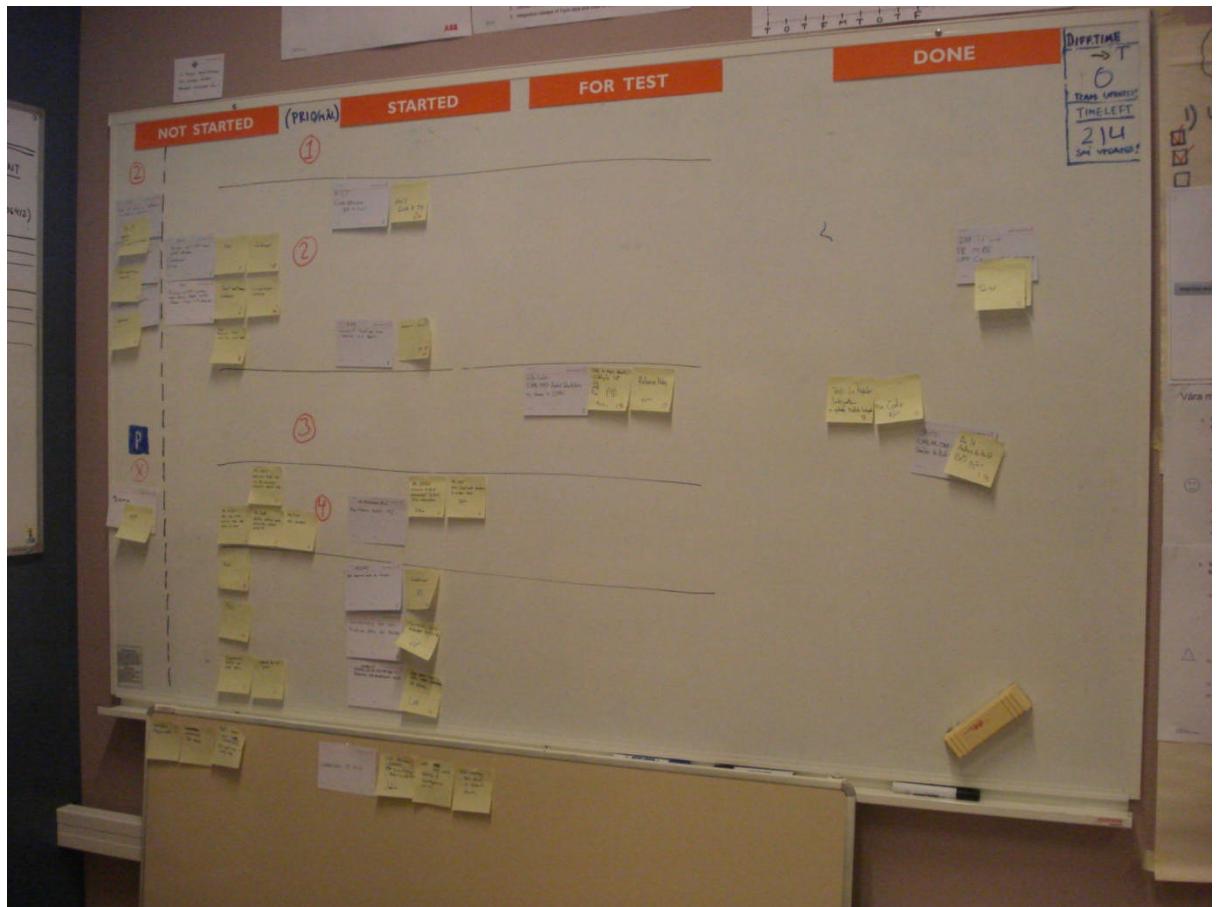


Figure 4.7.3-1 Scrum Board (Case C2)

In C3 the researcher attended and observed a number of agile practices. The Development Manager initially introduced the researcher to the team at a daily scrum meeting that the researcher attended. The team were informed that the researcher would observe the team during a number of agile practices. The researcher either stood or sat at the side of the room to observe the team and how they interacted. Similar to the previous two cases, notes were made on observations of the work environment and the interaction amongst team members. In all three cases the researchers personal thoughts such as speculations, or impressions were documented. Informal conversations also took place in this team during a lunch break or a coffee break. Similar to C2 observations or comments made by team members were documented and subsequently coded using NVivo See section 4.8.2 on p.131 for further detail.

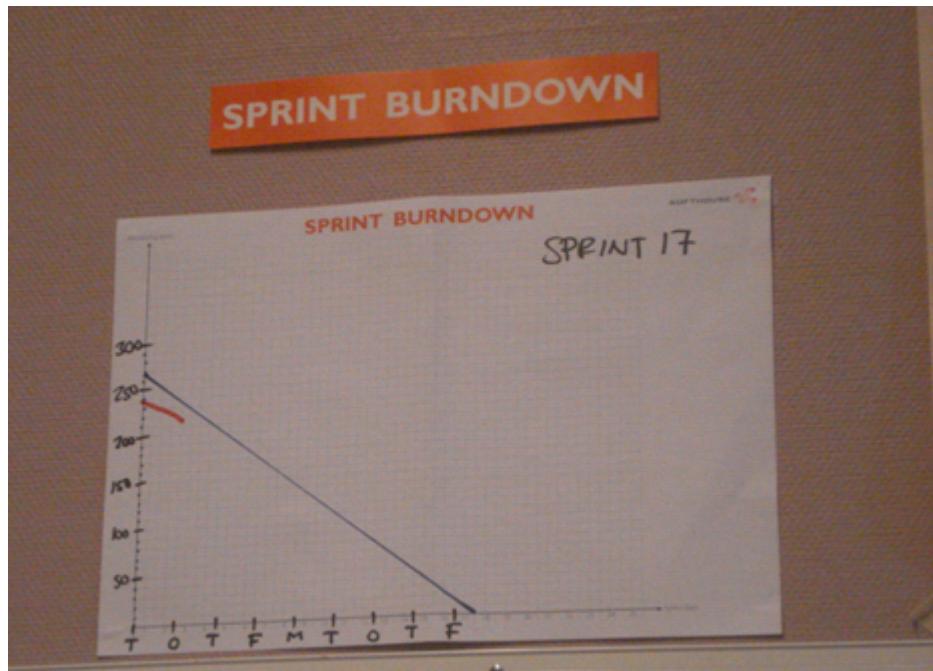


Figure 4.7.3-2 Burndown Chart (Case C2)

There were a number of difficulties with direct observation that the researcher needed to consider. It was difficult to determine exactly which actions were significant and which were not. At the outset the researcher attempted to capture as much detail as possible in each observation to try and avoid missing something important. As the number of observations increased and the researcher became more familiar with the practice being observed it became easier to identify what was different between observations, what was routine, and what had activities not been captured. In between each observation the researcher had time to reflect on what had been observed and consider what, if anything, the researcher needed to focus on in the next observation. As the number of observations of a particular agile practice were limited to one in some cases it was not always possible to accomplish this. In addition, the informal conversations held with various team members at different points throughout the day in C2 and C3 were also used to clarify the meaning of certain events or behaviours to ensure that the researcher did not assign a particular (incorrect) meaning to an event as recommended by (Corbin and Strauss, 2008, p30).

4.8 The Process of Analysing and Coding the Empirical Data

This section will first detail the coding process used in this study. It will then present the steps taken to analyse and interpret the qualitative data gathered.

4.8.1 Data Analysis

Analysis requires the examination and categorisation of data collected, the development of knowledge in relation to various aspects of the data collected, the interpretation of data, and the drawing of conclusions based on the empirical evidence available (Corbin and Strauss, 2008, p46; Lillis, 1999; Yin, 2009, p126). Some view the analysis phase as completely separate to the data collection phase, whereas others consider both phases as very closely linked with one informing the other (Myers, 2009, p166). Many researchers, particularly inexperienced qualitative researchers often find it difficult to analyse large volumes of qualitative data that has been gathered, which can include interview transcripts, field notes, and documents (Yin, 2009, p127). There is no standard structure for analysing the volumes of data generated by the case method and there is an absence of established techniques to ensure that qualitative data analysis is complete and impartial (Eisenhardt, 1989; Lillis, 1999). However, the data must be analysed using some qualitative data analysis approach that will help the researcher to focus on the important aspects of the data (Myers, 2009, p166). There are a number of different possibilities for analysing qualitative data, such as coding (Myers, 2009, p175). This technique is used by Miles and Huberman (1994) when conducting preliminary data analysis and is used in the preliminary analysis of this study

4.8.2 Coding

One of the easiest ways to analyse qualitative data is through the use of coding, where a code summarises and attaches meaning a group of words, a sentence or a paragraph (Miles and Huberman, 1994, p56; Myers, 2009, p167). Each code represents a concept which is derived from asking questions about the data, or making comparisons between data (Corbin and Strauss, 2008, p66). Coding data, means that the data is being analysed, which helps to reduce the size of the data and also assists with organising the data into themes or categories (Myers, 2009, p167).

There was some level of overlap between data collection and data analysis for this study, but this was minimal with the majority of the data analysis completed once the data collection concluded. As the intention in this study was to examine the empirical evidence in detail to try and generate new knowledge (Corbin and Strauss, 2008, p51) and a deeper understanding of clan control in ASD teams the data analysis was conducted in a systematic and comprehensive manner.

4.8.3 Coding process used in this study

A multistep analysis process was used to analyse the empirical data gathered, which was conducted as follows:

Step 1: The interview transcripts were read several times by the researcher to become immersed in the data, which according to Eisenhardt (1989) is an important step in the analysis process.

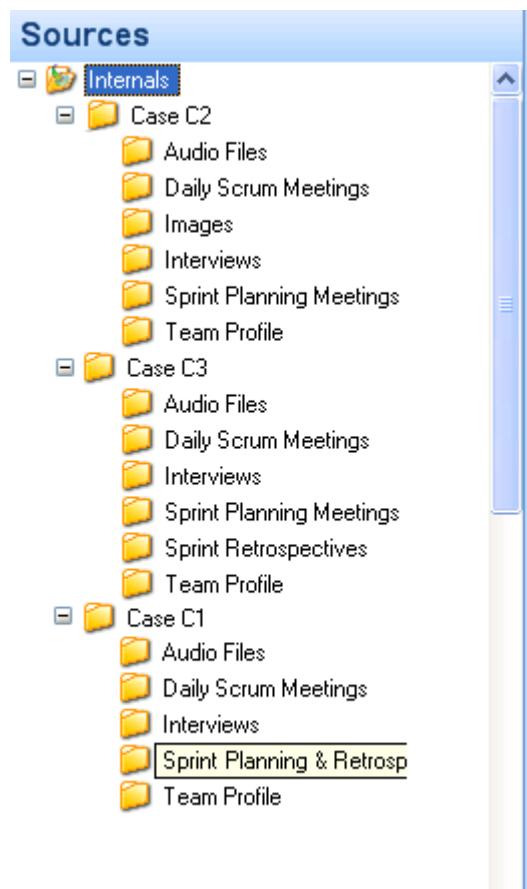
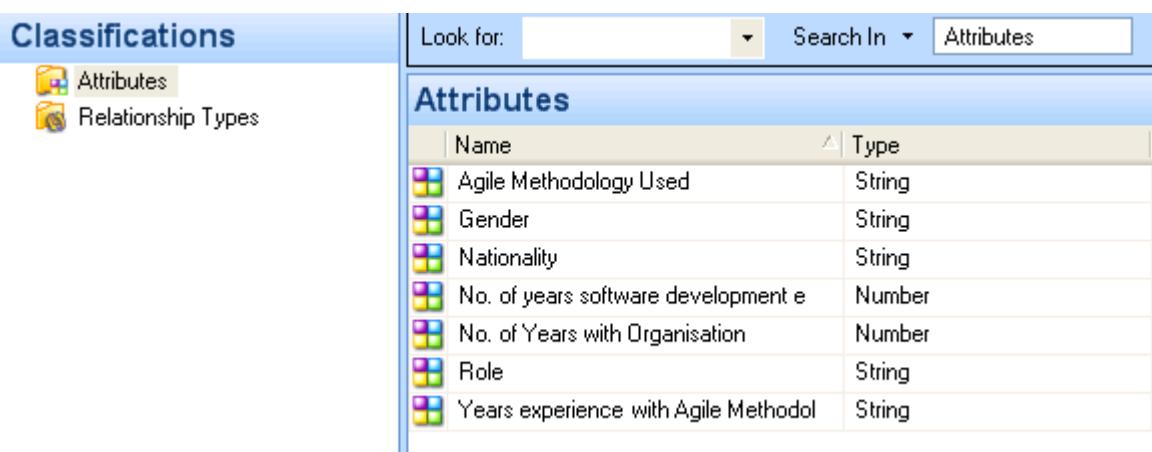


Figure 4.8.3-1 Data collected grouped by team in NVivo

Step 2: A case study database was created using NVivo (software for qualitative research), which comprised the interview transcripts, field notes, audio files and images grouped by team (Figure 4.8.3-1).

Step 3: A number of attributes were detailed in the case study database (Figure 4.8.3-2), which were populated with values; for example, the Gender attribute was assigned two values – male and female. Each participant ‘case’ was populated with values for each attribute. Using attributes allowed for the integration of some quantitative data to the existing qualitative data, which assisted with the interpretation and analysis of data in section 7.2 (p.268). For example, persons with more years experience may have had different views to those with less experience.



The screenshot shows the NVivo software interface. On the left, there is a sidebar with a blue header labeled 'Classifications'. Below the header are two buttons: 'Attributes' (highlighted in grey) and 'Relationship Types'. The main area has a light blue header labeled 'Attributes'. At the top right of this header are three buttons: 'Look for:', 'Search In ▾', and 'Attributes'. The main content area is a table titled 'Attributes' with a light blue header row. The columns are 'Name' and 'Type'. There are seven rows of data:

Name	Type
Agile Methodology Used	String
Gender	String
Nationality	String
No. of years software development e	Number
No. of Years with Organisation	Number
Role	String
Years experience with Agile Methodolo	String

Figure 4.8.3-2 Attributes defined for case study analysis

Step 4: The completed interview transcripts were formatted so that preliminary coding could be completed automatically by NVivo. As each interview had followed a structured interview protocol that contained the same sub-headings (Appendix A) it was possible to do this once the sub-headings in the interview transcripts were correctly formatted for NVivo i.e. all sub-headings had exactly the same name and were assigned a heading style of Heading 1. These sub-headings correlated to the resultant characteristics of clan control detailed in the conceptual framework in Figure 3.5-1 (p.77).

Step 5: Following the first round of auto-coding in NVivo each transcript was further examined to identify overlaps across the topics discussed in a second round of coding. On occasion an interviewee answered questions related to a particular topic, but the conversation also overlapped with another topic (for example; commitment and loyalty). Each interview was examined a second time to ensure that there were no oversights in relation to the coding. This was a lengthy process due to the number of interviews conducted.

Step 6: This next round of coding first reviewed the literature for each resultant characteristic of clan control to determine the factors that explained each characteristic as detailed in section 2.5.4 (p.31). These factors were mainly detailed as prompts in the interview protocol. The data already coded for each resultant characteristic of clan control was re-coded by factor. This step was repeated to ensure there was no oversight in relation to the identification of data for each factor.

Step 7: The fourth round of coding involved coding the fields notes gathered during the observations of the agile practices. The field notes for each observation were read to identify evidence of each characteristic of clan control. Where a segment of text could be interpreted to relate to a characteristic(s) it was coded under that characteristic(s). The field notes were re-read twice to ensure that none of the characteristics had been overlooked in the coding process. This resulted in the transition of some of the text coded to a different characteristic as it was deemed more appropriate. In some instances a section of coded text was removed from a characteristic as after reflection it did not relate specifically to an agile practice. For example, some data was initially coded under '*commitment*' where a quote indicated commitment to the team, but on later reflection this commitment did not manifest as a consequence of using one of the three agile practices studied.

Step 8: Once the field notes were coded a further (fifth) round of coding was required. The purpose of this round was to analyse the data coded under each factor by agile practice (Figure B-1 in Appendix B for sample coding). Once completed a period of time was allowed to elapse before examining the data again. The purpose of this was to allow time for reflection and to review the data from a fresh perspective and to ensure it had not been miscoded or misinterpreted during the

initial round of coding by agile practice. Further examination of the data resulted in the removal of some data from a characteristic, that was no longer considered appropriate and the addition of other data that was omitted in previous rounds of coding.

Step 9: Once the coding was completed a comparison was made across the three cases studied using each of the characteristics in the conceptual framework in an attempt to identify similarities and differences between different cases. This was completed at the level of agile practice by factor, which then answered the second research objective. For example, the data from each case was first compared on how the sprint planning meeting helped the teams studied to develop a sense of identity by examining how the sprint planning meeting helped members to feel part of the team, helped team members to have similar goals etc. This was followed by a comparison of how the daily scrum helped the three teams studied to develop a sense of identity and so on. A summary table was created for each combination of agile practice and characteristic from the conceptual framework, which detailed evidence of how each agile practice (Table 5.1.1-1 through to Table 5.3.7-1) facilitated the characteristic of clan control through the factors identified in the literature.

Step 10: Finally, the third research objective was answered by examining the findings from the three teams and identifying recommendations on how each agile practice may be improved to achieve higher levels of clan control (Chapter 6). Conclusions were drawn based on the interpretation of the findings, which are discussed in section 7.2 (p.268).

4.8.4 Write-up

The final stage of the analysis was the write up where the data gathered was presented and organised in a systematic way across the three cases. The write-up is presented in Chapter 5 and Chapter 6 with the conclusions drawn available in Chapter 7.

4.9 Summary

This study adopted an interpretive approach due to the exploratory nature of the research. Following consideration of the different research approaches and data

collection methods available and their various strengths and weaknesses the study conducted was qualitative and consisted of multiple-case studies. Two forms of data collection were utilised, of which interviews were the primary method for gathering evidence with direct observations used as supplementary evidence. The selection of cases was purposeful and each case was required to meet specific criteria in order to participate in the study. The research instrument was designed based on the conceptual framework (Figure 3.5-1), which was then tested in a pilot study. The remainder of this chapter addressed the approach used in this research to code and analyse the empirical data collected. Having provided all of this information Chapter 5 will present the results of this analysis and how they answer the research objective and Chapter 6 presents the recommendations on how each of the agile practices can help to achieve greater levels of clan control.

5 Findings and Analysis

5.0 Introduction

In this chapter the findings of the three case studies are presented and compared with the aim of answering the second research objective. It uses the conceptual framework (see Figure 5.0-1) to investigate how agile practices help ASD teams to achieve the resultant characteristics of clan control. Even though team members were interviewed individually a collective view of the team was presented as the team was the unit of analysis selected for this study (section 4.4.2, p.93). The intention was to go beyond the analysis of the agile practices in each single case to try and reveal any underlying causes or rationalization as to how these agile practices facilitate clan control.

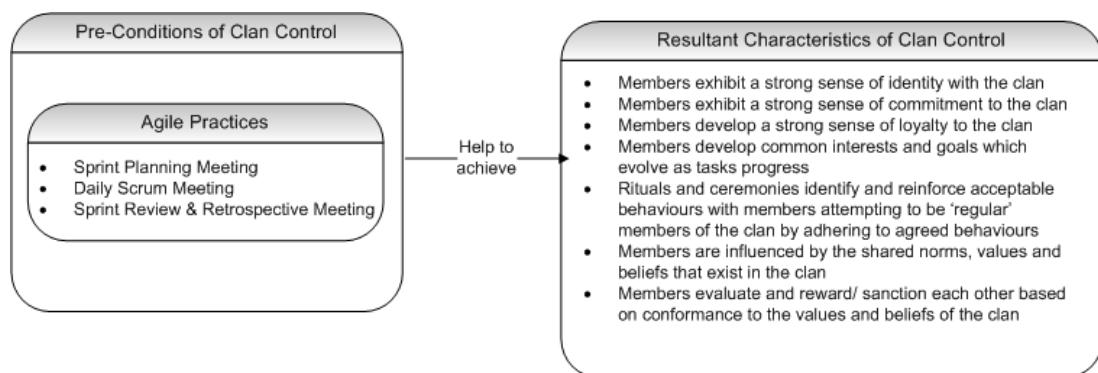


Figure 5.0-1 Conceptual framework used for this study

The structure of this chapter is as follows. The three agile practices are examined in turn under each characteristic of the conceptual framework in Figure 5.0-1. A number of the characteristics are very closely linked, as detailed in section 2.5.4 (p.31), resulting in some overlap. This adds an additional complexity to the analysis of the data as some characteristics are also presented as a factor for another characteristic; for example, loyalty and identity. Where this occurs this is acknowledged and the findings are presented in one place with the reader referred to that location in subsequent sections in order to avoid repetition.

The findings from each agile practice and characteristic of clan control are summarised in several tables from Table 5.1.1-1 through to Table 5.3.7-1. Each characteristic is illustrated with supporting evidence from the empirical data gathered through the use of interview data or field notes from observations. Where quotes are used each quote is attributed to the person who made the statement by using a naming convention such as Developer, Scrum Master, Product Owner, and so on. Where multiple persons hold the same role in a team; for example, developers, each is assigned a numeric identifier such as Developer 1, Developer 2, where Developer 1 is the first developer interviewed in a team. Observations are denoted as Obs C1, Obs C2 etc. where C1, C2 represent the cases studied. Where the responses of some participants contradicted those of other participants, both accounts are presented as suggested by Pettigrew (1990). Attempts are made to try and uncover the rationale or to identify evidence for the conflict when arriving at a conclusion. The chapter concludes with a summary of the findings.

5.1 Sprint Planning Meeting

This section discusses each of the resultant characteristics of clan control as detailed in Figure 5.0-1 (p.137) and illustrates how the sprint planning meeting (SPM) accomplishes or fails to accomplish each. It also uses the factors of each resultant characteristic identified in the literature (section 2.5.4 on p.31) as a lens to structure the analysis and the presentation of the findings in each of the tables. To recap what was stated in section 3.3.2 (p.66), the SPM is held at the start of each sprint where team members plan and agree the scope for the sprint in conjunction with the customer and estimations of effort for each task are determined and agreed.

5.1.1 Members exhibit a strong sense of identity with the clan

In section 2.5.4 (p.31) identification was defined as “*a psychological state wherein an individual perceives himself or herself to be part of a larger whole such as a work group, a team, or an organisation*” (Rousseau, 1998). Where individuals identify with a group, or they feel part of the team, are loyal to the team, have similar goals, values and norms and are committed to following them, cooperate and work as a team, and self-interest decreases in favour of team interests. These are the five factors that are examined in this section.

All team members reported that they felt part of the team with the exception of the Product Owners in both C2 and C3 who did not always feel part of the team. This appeared to be related to their role on the team “*sometimes it feels like the team see me as the other part - their boss*” (Product Owner, C2), or I feel like “*an outsider looking into the team*” (Product Owner, C3) even though both Product Owners regularly participated in the agile practices. The Product Owner in C2 was employed for 17 years by the organisation and had always worked in a technical role as part of a development team. With the introduction of Scrum he was asked to assume the Product Owner role by his manager. The team consequently changed their view of him as a developer and considered him a “*proxy/arbitrator between "us [the team] and them [the final customer]"*” (Developer 6, C2) rather than part of the team. In this case the team was considered as “*mainly the developers and Scrum Master*” (Developer 2, C2). This may explain why the Product Owner did not always feel part of the team. Yet, at the same time he did feel part of the team as he had long-standing relationships with team members through participation in several projects over a long period of time. In C3 the Product Owner was employed by the organisation for four years and he was only recently assigned as Product Owner to the project team studied, which may explain why he did not feel part of the team. He considered himself “*part of the product management team*” (Product Owner, C3) rather than part of the project team. Prior to his role as Product Owner he was a “*Solutions Architect, which gave me a business role in the team*” (Product Owner, C3). The remainder of the team also did not consider the Product Owner part of the team “*it is quite hard to say he [Product Owner] is part of the team...his involvement in the team is only maybe tenths of his time*” (Developer 1, C3). However, in both C2 and C3 there were conflicting opinions from team members as to who was part of the team. Some felt that everyone was “*part of the team...developers, QA, Product Owner, Scrum Master*” (Developer 3, C3), others felt it was just “*the workers, without Product Owner and Scrum Master*” (Developer 5, C2), others thought it was “*the developers and Scrum Master*” (Developer 6, C2, Developer 4, C3), or those who “*actually do the day to day work*” (Scrum Master, C3). These conflicting views were not evident in C1, which may be due to the length of time that agile was in use by this team, but it may also be because the Product Owner or Scrum Master roles did not exist in this team (Table 4.6.4-1 on page 115).

The roles in C1 were more akin to those of a traditional approach to software development and appeared to be clearly defined.

Members feel part of the team

The SPM helped team members develop a sense of identity within the team in a number of ways (Table 5.1.1-1) based on the factors identified in the literature (Table 2.5.4-1 p.34). The SPM was a frequent touch point for each of the three teams and helped team members to feel part of their team. Team members convened at the SPM at the start of each sprint to plan the tasks for the sprint with each team member given an equal opportunity to speak (Obs C1, C2, C3). This regular communication and interaction amongst team members helped them to feel part of the team, which was established by attendance at the meeting, participation in the meeting when expected, and the provision of input in addition to their expected contribution. Team members participated in “*a lot of discussion about tasks and how they should be solved and how they related to each other*” (Developer 4, C2). Where there were “*too many unknowns [about a task] very often it [the task] would be deferred in a planning session to an informal white-boarding session afterwards*” (Developer 4, C3). While team members in C1 did not always participate in the meeting (Obs C1), it did not mean that they did not feel part of the team. Their occasional absence was reported by team members present at the meeting as due to meeting conflicts with other departments/personnel within the organisation, often based in the United States. As the SPM occurred in the afternoon (Irish time) to facilitate the participation of the American distributed team member, it occasionally caused meeting conflicts due to the short timeframe in which meetings or communications can be conducted each day between Irish and American personnel.

C1 used the SPM to help new or distributed team members feel part of the team. Once a person “*joins the team they are expected to contribute to the sprint planning piece.*” (Developer 3, C1, Business Analyst, C1). This was particularly important for this team as the QA team members, based in India had changed frequently since the inception of the project “*In the start QA was in the States and then one guy left and another guy. Now, we've had QA resources in India for a year and one of them has just left.*” (Developer 1, C1). In C2 a new team member was quickly made part of the

team in his first SPM (Obs C2). This person joined the team on the first day the researcher was present observing the team. As this person had no prior knowledge of the project he was given a task to plan called '*Learning to Fly*', which was not related to the current project. He contributed at various stages throughout the SPM including the planning poker and estimation session where he had to identify sub-tasks and generate estimates for his assigned task (Obs C2). Consequently, he immediately had an opportunity to feel part of the team. C3 did not have any new team members since the introduction of Scrum, nor did it contain distributed team members.

The contribution of individuals in C1 to the SPM was sometimes haphazard (Obs C1). This did not mean that they did not contribute, but they only contributed when required and disengaged from the meeting when other discussions took place (Obs C1). Team members may have become complacent as a consequence of long-term participation in the SPM's (two years). However, the lack of engagement may also have related to the allocation of tasks to team members as each individual in this team had a main specialisation/role and they did not necessarily feel the need to contribute to a discussion when it did not affect their workload. They also had a tendency to disengage when distributed team members were speaking on the phone (Obs C1). This was partially due to the inability to clearly hear distributed team members due to the poor quality of the phone line, and also the difficulty with comprehending what was said due to language differences (Obs C1). In both C2 and C3, where all team members were co-located, team members engaged in discussion in addition to that which was expected (Obs C2, C3). Their regular contribution may be due to the recent introduction of Scrum and their desire to engage with the practice. It may also be related to their close proximity to one another. However, it is possible that over time their level of engagement may decrease as they become more familiar with the practice and with the technical aspects of the project where less discussion is required.

Members are loyal to the team

The SPM provided evidence that team members are loyal to the team. See section 5.1.3 (p.156) for a detailed discussion on loyalty.

Team members have similar goals, values, and norms and are committed to following them

The SPM helped teams to develop similar goals, values and norms. Attendance at the SPM in C2 and C3 illustrated their commitment to the goals, values, and norms of the team by contributing to discussions, sharing information, learning, communicating, and negotiating requirements with the Product Owner (Obs C2, C3), although team members in C1 did not always participate in the meeting (Obs C1). In C1 and C2 participation of new or distributed team member in the SPM helped them to become familiar with the norms and values of the team. As discussed earlier a new team member in C2 immediately had an opportunity to become familiar with the norms and values of the team as he participated in and contributed to the SPM on his first day of employment with the organisation (Obs C2). C3 did not have any new team members since the introduction of Scrum, nor did it contain distributed team members.

The SPM ensured that a common goal was established and was clearly defined for each sprint. The common short-term goal for the sprint was recorded in the Product Backlog, which was available to all team members ensuring that all team members had clarity on the goal (Obs C1, C2, C3). In C1 the SPM ensured that the goals of distributed team members were aligned with the goals of the QA team in India by communicating with them “*every ten days [from a planning perspective]*”. Also, during the SPM the Project Manager reinforced the goals of the team to QA. This was important due to cultural differences as the Indian team tended to be hesitant and appeared not to want “*to disturb... it is almost like I would say afraid ... because we know the stuff and they're trying to learn it.*” (Developer 2, C1), or “*they tend to give you a more positive picture of things when we actually start doing detailed sprint planning*” (Project Manager, C1).

The SPM also provided teams ownership over the team goals and their personal goals. In all three cases responsibility was allocated to the team to define and clarify the goals for each sprint and to determine how they were achieved. All team members were involved in planning tasks for each sprint with the teams “*prioritising*

tasks" (Project Manager, C1), deciding "*which tasks will be in the sprint*" (Developer 1, C3), "*what priorities the stories are*" (Developer 2, C1), "*how many points we think are reasonable to execute in a sprint*" (Developer 1, C2), "*who is going to do them [the tasks]*" (Developer 2, C1) and "*what we have to do in order to feel secure about the quality.*" (Developer 1, C2). In each of the three teams the goal was seen as a "*team goal*" rather than an individual goal "*so, if someone is straying then it means that the team has a harder time*" (Developer 1, C2), or "*you are letting the team down*" (Developer 4, C3). Likewise, if an individual in the team struggled with a task or "*if something goes wrong, even if it's in a specific area that someone is responsible for, it would be very much a team effort to fix it*" (Developer 5, C1).

The SPM provided teams with autonomy to allocate tasks in whatever way they deemed appropriate. In C1 where the team were using an agile methodology for two years (Table 4.6.4-1) tasks were assigned independently of the SPM by the Project Manager to "*whoever is responsible for a certain area*" (Developer 2, C1). This behaviour had emerged as when "*you are under pressure ... people who know certain areas will be thrown at that area again.*" (Developer 1, C1). This was not unexpected in this team as this team was recently under severe pressure to deliver functionality in a short period of time. It was natural in a pressurised situation to allocate work to those who can complete it in the shortest timeframe in order to achieve a deadline. In C3 tasks were assigned through a collaborative process at the SPM (Obs C3).

However, an unusual norm had emerged in C2 in relation to the assignment of tasks, as a consequence of their newly acquired autonomy, which they embraced. Recently, "*when it comes to sprint planning they have changed to working in pairs, or threesomes.... it is good and bad because it is good to see that they are transferring knowledge, they are sharing, they are I would say solving some issues much faster when they are at least two.., but productivity has gone down by half.*" (Product Owner, C2). This was something that emerged within the team, yet it seemed acceptable to the team to continue like this even though the lower productivity was highlighted to the team by the Product Owner in the SPM (Obs C2). The teams in C1 or C3 did not exhibit this trait with all team members' assigned individual tasks, which were recorded in the Product Backlog (Obs C1, Obs C3). This may have

emerged in C2 for a number of reasons. Prior to the introduction of Scrum, team members had expressed a desire to “*to spread the knowledge*” (Product Owner, C2) across team members, which was not facilitated. With Scrum, the team realised they had “*the freedom to select how they implement stuff on their own*” (Product Owner, C2) and they began to allocate two or three team members to each task. It may also be because the team had difficulty estimating tasks due to their unknown or complex nature, and it was possible that they began to work in pairs in an attempt to complete tasks. It may also have emerged due to their desire to “*spread the knowledge instead of being one person that can do this thing and no-one else knows it*” (Product Owner, C2). Even though this behaviour had emerged in the team one developer felt that there as “*not as much as knowledge cross-over as there should be ...it's all you know 'get it done as quickly as possible'*” (Developer 2, C2). This view was held by one of the least experienced members of the team, who was employed by the organisation for less than three years. It was possible that his desire to learn was greater than those who were with the organisation substantially longer.

In C1 a different norm had emerged in relation to the team’s autonomy over personal goals. As team members became more experienced they expressed a desire to work on a particular task as “*on an individual level, people want the best areas of work where they would learn the most.*” (Developer 4, C1), or they may “*want to up their profile, to progress their own careers*” (Business Analyst, C1). The SPM provided a forum where such requests could be made and accommodated. Similar experiences were not found in C2 or C3. Instead, in C2 the team “*have sacrificed time, from time to time in order for a new person to start doing a task that hasn't been done before*” (Developer 6, C2), or in C3 the team “*try to avoid always the same people always doing the same kind of task. We try to share it around*” (Developer 5, C3), whereas in C1 “*tasks are generally assigned to people who can do them the quickest.*” (Developer 2, C1). This may be a consequence of the pressure this team faced to deliver a large amount of functionality in a short timeframe where tasks needed to be completed quickly. The difference between the team in C1 and the other two teams may also be attributed to the redundancies that took place in the previous year in this organisation as staff may have a desire to continuously learn to keep up to date with new technologies to ensure their continued or future employability.

The SPM also required the participation of the customer to ensure that their goals were aligned with the goals of the remainder of the team. In C2 and C2 this goal was agreed in conjunction with the Product Owner, who always participated in the SPM and worked with the team to define the goals for the sprint, and to “*prioritise items*” (Developer 3, C2). It was “*obvious for everyone what we will do for two weeks*” (Product Owner, C2). On occasion the team negotiated with the Product Owner to “*remove things from the scope*” (Developer 2, C2), “*to postpone or to bring other tasks if needed in a sprint*” (Developer 1, C3) if the team felt that amount of work expected for the sprint was too much, or if additional work needed to be added to the sprint. The customer rarely participated in the SPM in C1 and was not considered “*an integral part of our team*” (Project Manager, C1). The customer in this team was distributed (based in the United States) and team members found it “*hard to get their [customer] time, ...they are very slow to make decisions... it might take a week to get a response [from the customer]*” (Developer 2, C1). This caused difficulty for the team as often the team had to “*assume things*” (Developer 1, C1) and it emerged at a later date that the customer “*want some things differently to what we have planned....and we have to change what we have done*” (Developer 3, C1).

In C1 and C3 QA participated in the SPM. As none of the QA personnel were interviewed in C1 it was only possible to present the views of the remainder of the team in relation to QA. In C1, the QA function was based in India, but they participated regularly in the SPM even though this required these team members to work additional hours on the days that the SPM was conducted (Obs C1). However, team members felt that the participation of QA was limited “*they give their status and stuff and then just go back and speak to the domestic team*” (Developer 1, C1), and also because “*they keep chopping and changing the [QA] people, so we've to go through the whole process again [explaining the project etc.]*” (Developer 2, C1). In C3, QA were involved from the outset, which was a dramatic change for the way in which the team and QA viewed each other “*QA would have limited their interaction altogether with them [development team] and now they would have no qualms. Everybody is talking to the developers and walking over to them and questioning them.*” (QA, C3). This was considered “*crucial*” (QA, C3) from a QA perspective as the SPM provided an opportunity to communicate the requirements, limitations and

commitments of QA to the team (Obs, C3), which resulted in a better understanding between the two groups and it also resulted in QA “*feeling part of the team*” (QA, C3). No QA staff existed on the team in C2.

Team members work and cooperate as a team

The SPM helped the three teams to work and cooperate together through collaboration to establish a common goal for the sprint. In C2 and C3 this was also contributed to by the collaborative action of assigning and estimating tasks, which was conducted using a technique called planning poker (Obs C2, Obs C3). Working together on such tasks helped team members to become more cooperative as a team, as they had to reach a common consensus for each estimate. At the outset teams were “*extremely poor at estimating, whereas now they are more accurate*” (Developer 2, C3). It was also observed in both C2 and C3 that in most cases developers played the same planning poker card when estimating tasks, which indicated that developers were converging towards a similar understanding of the amount of effort required to complete particular tasks. It is possible that the use of planning poker may diminish over time as the teams become more familiar with the project and the technology, resulting in similar estimates that can be verbalised without the need to use the cards. This contrasted with C1 where tasks were assigned independently of the SPM by the Project Manager to “*whoever is responsible for a certain area*” (Developer 2, C1), that were reviewed and clarified in the SPM (Obs C1). In addition, each developer defined “*their own estimates*” for each task, which were conducted independent of other developers (Developer 1, C1). This may be because this many of the team were very experienced and familiar with the architecture of the project, which had been in existence for two years (Table 4.6.4-1). It may also be because team members had developed specialisation in particular tasks over time and were predominantly assigned those tasks as part of their workload, and therefore were in the best position to determine the most accurate estimate for the task.

The SPM also promoted cooperation between the Product Owner and the remainder of the team and helped them to understand each others needs. In C2 and C3 the Product Owner was considered a valuable part of the team as they prioritised tasks on the Product Backlog for the team (Obs C2, C3) and the team “*can ask him*

[questions] and get instant feedback on decisions" (Scrum Master, C3). In C1 the customer was not involved in any collaborative interaction with the team (Obs C1) and as already stated was not considered part of this team. The customer did not participate for a number of reasons, which were reported by the team. One of the reasons for the lack of customer participation related to the software under development as it was a "*very technical back-end web services. There is no front-end*". *If you put the customer in front of us I doubt they would understand*" (Developer 1, C1). Also, as the project was so large there was no one single defined customer. Instead, there were "*a few different people in a few different areas...with no one person who understands it all*" (Developer 1, C1). It is possible that lack of participation was also contributed to by their location as the customer was based in the United States, whereas the core development team was based in Ireland.

Team interests emphasised over self-interests

Finally, while it was not possible to determine if self-interests decreased over time as a longitudinal study was not conducted there was some evidence to suggest that team members placed emphasis on the interests of the team rather than their own personal interests. The SPM helped to develop a strong team ethos within each team ensuring that the team worked together to deliver on their goals. In C1 this was demonstrated by the Project Manager assigning tasks to "*whoever is responsible for a certain area*" (Developer 2, C1), which suggested that the Project Manager assigned the most experienced or appropriate person to each task in the interests of the team delivering a good quality product. However, it could also be suggested that the Project Manager was protecting his own interests to ensure that his team delivered as expected to avoid any negative repercussions from more senior personnel to which he reported. No complaints to this approach were reported by the team even though "*it leads to lack of knowledge of the whole project throughout the team*" (Developer 1, C1), but it was recounted that if team members felt "*strongly about something you'd definitely get onto it if you liked*" (Developer 2, C1).

In C2 team members had a desire to share information and knowledge throughout the team, which was achieved by rotating tasks amongst team members "*we can discuss if someone else is going to do the work, so we have information spread*"

(Developer 6, C2). According to the Product Owner, the team have repeatedly “cried out for this [opportunity to share knowledge] for several years” (Product Owner, C2). This could demonstrate some level of self-interest on the part of team members to increase their own knowledge, but it could also be interpreted as considering the best interests of the team to ensure that knowledge is shared amongst team members and that no single person possesses all the knowledge in a particular area. However, on occasion, similar to C1 they “try and find the guy that will do it [the task] the best and in the shortest time span” (Developer 6, C2) if it is required, but they will also allow time “for a new person to start doing a task that they haven’t done before.” (Developer 6, C2). The team in C3 also tried to share knowledge and gain different experience by trying “to avoid always having the same people doing the same kind of task. We try to share it around.” (Developer 5, C3), but “if somebody is behind in tasks somebody else will step in and help” (Scrum Master, C3). This suggested that the team wanted to complete tasks for the good of the team, but it may also be to portray a good image to management that the team were delivering as promised. The Scrum Master reported that while team members were allowed to indicate their preference for a task, they rarely did, which illustrated little self-interest on the part of team members as often “everybody is waiting for the other to choose something.” (Scrum Master, C3).

Characteristic of Clan Control: Members exhibit a strong sense of identity with the clan			
Factor	Case C1	Case C2	Case C3
Members feel part of the team	✓ SPM is a frequent touch point for the team	✓ SPM is a frequent touch point for the team	✓ SPM is a frequent touch point for the team
	X SPM provides an opportunity for team members to communicate and interact regularly, which does not always occur	✓ SPM provides an opportunity for team members to communicate and interact regularly	✓ SPM provides an opportunity for team members to communicate and interact regularly
	✓ SPM provides an opportunity for new and distributed team members to regularly participate in the team	✓ SPM provides an opportunity for new team members to regularly participate in the team	No new or distributed team members
Members are loyal to the team	See Table 5.1.3-1, p.161		

Characteristic of Clan Control: Members exhibit a strong sense of identity with the clan			
Factor	Case C1	Case C2	Case C3
Team members have similar goals, values, and norms and are committed to following them	X SPM requires regular attendance at the meeting to become familiar with norms and values, which does not always occur	✓ SPM requires regular attendance at the meeting to become familiar with norms and values	✓ SPM requires regular attendance at the meeting to become familiar with norms and values
	X SPM provides an opportunity for team members to contribute in addition to their expected contribution, which does not always occur	✓ SPM provides an opportunity for team members to contribute in addition to their expected contribution	✓ SPM provides an opportunity for team members to contribute in addition to their expected contribution
	✓ SPM establishes a common set of short-term goals for the sprint	✓ SPM establishes a common set of goals for the sprint	✓ SPM establishes a common set of short-term goals for the sprint
	✓ SPM provides clarity on short-term goals set e.g. creation of Sprint Backlog	✓ SPM provides clarity on short-term goals set e.g. creation of Sprint Backlog	✓ SPM provides clarity on short-term goals set e.g. creation of Sprint Backlog
	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of each task to one team members	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of tasks to two or more team members	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of each task to one team members
	✓ SPM provides individual team members autonomy over their personal goals e.g. team members can express a desire to work on particular tasks	✓ SPM provides individual team members autonomy over their personal goals e.g. team members share and rotate tasks	✓ SPM provides individual team members autonomy over their personal goals e.g. team members share and rotate tasks
	X SPM requires participation of the customer to ensure their goals and the team goals are aligned, but the customer rarely participates	✓ SPM requires participation of the Product Owner to ensure their goals and the team goals are aligned	✓ SPM requires participation of the Product Owner to ensure their goals and the team goals are aligned
	X SPM provides an opportunity for QA to participate in and contribute to the SPM but QA may limit their contribution	No QA present in the team	✓ SPM provides an opportunity for QA to participate in and contribute to the SPM
	✓ SPM involves all team members in the establishment of common goals for the sprint	✓ SPM involves all team members in the establishment of common goals for the sprint	✓ SPM involves all team members in the establishment of common goals for the sprint
Team members work and cooperate as a team			

Characteristic of Clan Control: Members exhibit a strong sense of identity with the clan			
Factor	Case C1	Case C2	Case C3
	X SPM involves all team members in tasks e.g. task estimation, task assignment	✓ SPM involves all team members in tasks e.g. task estimation, task assignment	✓ SPM involves all team members in tasks e.g. task estimation, task assignment
	X SPM promotes cooperation between the customer and the remainder of the team, but the customer rarely participates	✓ SPM promotes cooperation between the Product Owner and the remainder of the team e.g. agreement on prioritisation of tasks	✓ SPM promotes cooperation between the Product Owner and the remainder of the team e.g. agreement on prioritisation of tasks
	✓ SPM promotes cooperation between QA and the remainder of the team	No QA present in the team	✓ SPM promotes cooperation between QA and the remainder of the team
Team interests emphasised over self-interests	✓ SPM helps the team to work together to deliver on their agreed goals for the sprint e.g. assigning tasks to the most appropriate person	✓ SPM helps the team to work together to deliver on their agreed goals for the sprint e.g. sharing information and rotating tasks	✓ SPM helps the team to work together to deliver on their agreed goals for the sprint e.g. sharing information and rotating tasks

Table 5.1.1-1 SPM: Members exhibit a strong sense of identity with the clan

5.1.2 Members exhibit a strong sense of commitment to the clan

Team commitment was described as where teams (clans) develop goals and values that members may accept, members exert considerable effort on behalf of the team (clan), members may exhibit a desire to remain a member of the team (clan), members identify with the team, and are loyal to the team (section 2.5.4, p.34). These are the five factors that are examined in this section.

The SPM contributed to the development of commitment to the team in a number of ways (Table 5.1.2-1). As discussed in section 3.3.2 (p.65), the SPM is a meeting held at the start of each sprint where teams plan their workload for the sprint.

Team develop goals and values that members accept

In all three cases the SPM helped teams to develop similar goals and values that team members may accept, which were detailed in section 5.1.1 (p.138). To assist with developing the common goal for the sprint tasks were broken down into “*smaller detail*” (Developer 1, C2), so “*you have a precise idea which features are going to be part of the release*” (Developer 5, C3) and “*you’ve a fair idea of what*

everyone on the team is doing, exactly what they're doing" (Developer 2, C1). In C2 and C3 tasks were prioritised in conjunction with the Product Owner whereas in C1 tasks were prioritised by the Project Manager as the customer rarely participated in the SPM. Each task was discussed in turn by the team with conflicts and trade-offs identified and estimates determined and agreed (Obs C1, C2, C3). A final decision was ultimately agreed by the team in relation to each task and a team member was assigned to each task, which was recorded in the Sprint Backlog and made available to all team members so that they had clarity on the goals for the sprint (Obs C1, C2, C3). Documenting this information in the Sprint Backlog was a written commitment by the team to accept the goal to deliver the agreed functionality by the end of the sprint with team members feeling that they now "*have a much better picture of what they have to do*" (Developer 2, C3).

Members exert considerable effort on behalf of the team

The SPM provided some evidence that team members were willing to expend considerable effort for the team by ensuring that they felt personally accountable for the tasks assigned "*you're responsible for your part, your work, your task*" (*Technical Architect, C1*) as well as feeling "*responsible for the performance of the team*" (Developer 7, C2). This may have occurred because team members felt part of the team and as described in section 5.1.1 (p.138) they believed that the goal was a team goal. The acceptance of tasks assigned demonstrated their commitment to the team to deliver what they have promised, "*you are not only making a commitment to the requirement you are also committing to the team, you have committed to fulfil that*" (Developer 4, C3) and "*if you take the task you feel responsible for it - to complete it*" (Scrum Master, C3). Early in the Scrum process in C3 team members underestimated tasks during the SPM and committed to completing the tasks in the timeframe indicated, which they then felt obligated to achieve as they had provided the original estimate. This often resulted in trying to "*cram it all in and not doing the best quality work that could have been done*" (Developer 2, C3). However, as the team became more familiar with Scrum and the SPM's they began to realise that they did not need to over commit themselves and became more comfortable with stating if "*it can't be done in two days. It will be more like – I have looked into it now and our estimate is completely off now and it is at least 3 days, maybe 4*"

(Developer 2, C3). This contrasted with C1 where if team members committed to an estimate they felt they could not renege on what was pledged to deliver “*There’s the feeling ‘I’ve said I’ll get this done in 5 days, I better get it done in 5 days’.* You are committing to it as opposed to someone else saying where you can always turn it around and say ‘I didn’t say I could get it done in 5 days’.” (Developer 1, C1).

However, the SPM did not prevent teams in C1 and C2 from incorporating too much functionality into a sprint, even though the team were aware of the availability of resources for the sprint, the amount of time available to them, and the estimates for tasks. This resulted in team members exerting substantial additional effort in order to deliver the functionality agreed at the start of the sprint. In C1 the team were under severe “*time pressure in the last number of months.... because management (I think) committed to this crazy deadline*” (Developer 1, C1), which resulted in the addition of more tasks to the Product Backlog and required regular overtime for the team for a number of months in order to achieve the sprint goals. Even though this time pressure had abated with the team now working “*normal weeks*” (Developer 1, C1) a number of developers were regularly willing to exceed the amount of hours required with many “*regularly doing 70 hour weeks so...they seem to want to. There’s no pressure put on them*” (Project Manager, C1), which in the view of the Project Manager could be caused by the agile practices as they “*definitely give more ownership to the developer*” (Project Manager, C1). It could be assumed that these developers were working this additional time for the benefit of the team as no monetary rewards existed for working overtime, but it could also be to portray an image to management that they were hard-working and dedicated to their job as there were recent “*redundancies, so people just want to keep the head down*” (Developer 1, C1).

In C2 “[management] put the pressure on that we have had to take more into the sprints than we have actually can do.” (Developer 3, C2). Consequently, the team included more tasks in the Sprint Backlog even though they felt the expectations were too high. They demonstrated their additional commitment to achieve the desired functionality by working overtime as “*several members of the team have been working overtime*” (Developer 3, C2). This contrasted with the view of the Product Owner who was not aware of this “*we haven’t really worked any overtime*

since we introduced Scrum” (Product Owner, C2). However, when informed that team members were working overtime the Product Owner indicated that there was an “*economic incentive to sit here and work late*” as developers received overtime payments. While this may be an incentive to work overtime team members reported that “*I have personally worked more overtime now since the start of Scrum*” (Developer 7, C2) and that “*the last 3 or 4 last sprints the pressure was very intense... and I worked overtime*” (Developer 3, C2). However, even with this additional commitment from the team to complete tasks the team have failed to deliver the functionality in several sprints “*since we took in more in the sprint than we could do then we failed in the tasks*” (Developer 3, C2). No similar evidence was reported in C3.

The level of effort expended by the Product Owners in C2 and C3 illustrated a commitment to the needs of the team above that normally expected. This was a new role for both Product Owners when Scrum was first introduced and “*now I literally do not stop thinking about the product, 24 hours day. When I became Product Owner, it just consumes you. I just dream about it now.*” (Product Owner, C3). Both Product Owners expended extra effort in order to be prepared for a Monday morning SPM “*the sprint planning on Monday means that me as Product Owner has to work Sunday evening to be prepared Monday morning.*” (Product Owner, C2). The SPM provided a “*deadline in that you can’t say “I’ll do that Tuesday”. It has to be Monday for the planning [meeting].*”(Product Owner, C3). In both cases this additional work was unpaid, but no complaints were made in relation to this from either Product Owner. Both were willing to expend the extra effort for the benefit of the team. The Product Owner role did not exist in C1 and therefore there this was not evident.

Member exhibit a desire to remain a member of the team

Sixteen of the 25 individuals interviewed indicated that they had a desire to remain part of the team. This was partially contributed to by the composition of the team as the team is considered “*a good team*” (Developer 3, C2) and the people on the team “*are good to work with*” (Business Analyst, C1). It was also partially due to the agile methodology itself, but could not be attributed to a single agile practice as team

members “*like the way this team work*” (Developer 5, C3) since the implementation of an agile methodology. Individuals had no desire to work on a team that “*don’t follow agile*” (QA, C3) with a “*preference for agile to the traditional approach*” (Developer 3, C1), but this was due to the agile methodology as a whole rather than specific agile practices. Reasons for possible departure of the nine remaining team members were given as “*it would be great to be on a project from the start*” (Developer 3, C1; Project Manager, C1), more challenging work “*if I would have more fun things to do*” (Developer 1, C2), working with “*different people*” (Developer 3, C3), seeking a change in work “*just because I have been here so long*” (Developer 2, C2), because “*I have grown tired of what I’m doing in this team technically*” (Developer 6, C2), or finally if a “*good opportunity*” (Developer 3, C3) came along. However, in C3 it was reported by the development manager in an informal conversation that one individual had departed the team since the introduction of Scrum due to the increased visibility that Scrum had brought to the team. This individual did not like the scrutiny and intrusion that Scrum brought to the team, so he had left the organisation.

Members identify with the team

The SPM contributed to team members developing a sense of identification with the team, which was discussed in section 5.1.1 on p.138.

Members are loyal to the team

The SPM also contributed to the development of a sense of loyalty to the team, which are discussed in greater detail in section 5.1.3 on p.156.

Characteristic of Clan Control: Members exhibit a strong sense of commitment to the clan			
Factor	Case C1	Case C2	Case C3
Team develop goals and values that members may accept	✓ SPM establishes a common set of short-term goals and values for the sprint	✓ SPM establishes a common set of short-term goals and values for the sprint	✓ SPM establishes a common set of short-term goals and values for the sprint
	✓ SPM provides clarity on short-term goals set e.g. creation of Sprint Backlog	✓ SPM provides clarity on short-term goals set e.g. creation of Sprint Backlog	✓ SPM provides clarity on short-term goals set e.g. creation of Sprint Backlog
	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of each task to one team members	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of tasks to two or more team members	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of each task to one team members
	✓ SPM provides individual team members autonomy over their personal goals e.g. team members can express a desire to work on particular tasks	✓ SPM provides individual team members autonomy over their personal goals e.g. team members share and rotate tasks	✓ SPM provides individual team members autonomy over their personal goals e.g. team members share and rotate tasks
	No customer present in the team	✓ SPM requires participation of the Product Owner to ensure their goals and the team goals are aligned	✓ SPM requires participation of the Product Owner to ensure their goals and the team goals are aligned
Members exert considerable effort on behalf of the team (clan)	✓ SPM ensures team members feel personally accountable for their tasks e.g. desire to complete tasks when the team have overcommitted on what can be achieved	✓ SPM ensures team members feel personally accountable for their tasks e.g. desire to complete tasks when the team have overcommitted on what can be achieved	✓ SPM ensures team members feel personally accountable for their tasks e.g. desire to complete tasks when the team have overcommitted on what can be achieved
	No customer present in the team	✓ Product Owner willing to work additional hours at the weekend to prepare for the SPM on Monday morning	✓ Product Owner willing to work additional hours at the weekend to prepare for the SPM on Monday morning
Exhibit a desire to remain a member of the team (clan)	Not evident	Not evident	Not evident
Members identify with the team	See Table 5.1.1-1, p.150		
Members are loyal to the team	See Table 5.1.3-1, p.161		

Table 5.1.2-1 SPM: Members exhibit a strong sense of commitment to the clan

5.1.3 Members develop a strong sense of loyalty towards the clan

Loyalty to an organisation was defined as “*a feeling of affection for and attachment to an organisation*” (Buchanan, 1974), which can also be applied in the context of a team or a group where an individual feels an affection for or an attachment to a team or a group (section 2.5.4, p.35). Individuals who are loyal to a clan or a team develop a feeling of affection for and attachment to the team, identify with the team, develop a sense of belonging and solidarity with the team, team interests are emphasised over self-interests, and they develop a strong team spirit. These are the five factors that are examined in this section.

The SPM contributed to the development of loyalty within the team in a number of ways (Table 5.1.3-1), although team members in all three cases reported that they had a strong sense of loyalty, but this was not specifically related to the SPM. One team member felt that loyalty “*is an attribute I have always had*” (QA, C3), or “*if I do my work properly, or well, it is because I like to do my work properly or well, not for the team*” (Developer 3, C3). The QA resource now felt that “*my loyalty to the team is beyond QA*” (QA, C3) and that she now felt part of a wider team effort, but this was not specifically related to one single agile practice. In C1 there were a number of individuals who were considered too loyal to the team by others as they continuously worked long hours when they were not required, and in the view of one developer they “*probably need to be told to go home*” (Developer 1, C1). This developer believed it is in their “*mindset*” to do this rather as they do not seek praise or recognition for it, but it may also relate to reasons other than loyalty to the team.

Team members develop a sense of attachment to the team and a sense of belonging and solidarity to the team.

The SPM facilitated regular communication, collaboration and interaction amongst team members including distributed team member, which helped team members to develop a sense of attachment to the team and a sense of belonging to the team. The SPM supported teamwork “*it is helping the team work more efficiently as a team, not as individuals*” (Scrum Master, C2) and it felt “*more like a group because you work with everybody in the team*” (Scrum Master, C3). Everything is “*a team effort ... you feel more loyal towards the team's tasks so to speak.*” (Developer 4, C2). At the

same time one of the least experienced developers in C1 felt that there was “*a certain degree of don’t really care about each other, they [the developers] are just doing a job*” (Developer 4, C1), even though he personally did feel “*part of the team*”. This view may be attributed to the “*fact that we [the team] got affected by a lay-off last year*” (Developer 4, C1) and he felt that everyone was looking out for themselves as a result. The Project Manager in this team felt that the SPM allowed team members to become “*more comfortable with each other, more confident with each other, and a friendship or some sort of human relationship builds up between them*” (Project Manager, C1), which helped team members to feel part of the team. This view point may be related to this person’s role in the team, which allowed him to observe the changes in the team over time. As discussed in section 5.1.1 (p.138) a lack of attendance at and contribution to the SPM was observed in C1 with team members disinterested and failed to contribute when a discussion related to a task that did not impact on their workload (Obs C1). However, this did not appear to impact on their sense of belonging or feeling of attachment to the team with everyone in this team reporting that they felt part of the team. The SPM also helped a distributed team member in C1 to “*feel part of the team*” (Technical Architect, C1), which he believed was partially contributed to by the “*continuous communication between the team*” (Technical Architect, C1) at the SPM. This was only evident in C1 as this was the only team studied that had distributed team members (Table 4.6.4-1, p.115).

The SPM facilitated regular customer participation, but in C1 there was no evidence that the customer was considered “*part of the team*” (Business Analyst, C1) by any of the team members interviewed. As the customer was rarely present at the SPM it was unlikely that they felt a sense of attachment to the team due to their lack of participation and interaction with the team. This contrasted with the regular participation of the customer representative (Product Owner) at the SPM in C2 and C3 (Obs C2, Obs C3). The team members in both C2 and C3 considered the Product Owner a valuable part of the team because they participated in the SPM and provided feedback to the team “*I think that is a major improvement that we have a Product Owner and we can ask him anytime and get instant feedback on decisions*” (Scrum Master, C3). However, as reported in section 5.1.1 (p.138) the Product

Owners themselves had conflicting views on whether they felt part of the team. The Product Owner in C2 indicated that “*sometimes yes [I feel part of the team], sometimes not [I don’t feel part of the team]*” (Product Owner, C2). He felt that when he puts “*up goals for the team*” the team “*see me as their boss*” (Product Owner, C2), yet when the team is delivering a release to a customer “*we are one unit delivering to someone... I get their support in what we deliver*” (Product Owner, C2). The Product Owner in C3 also felt that his loyalties were divided as he was representing the customer when participating in the SPM “*I take their [the customer] input and go to the development team and I drive whatever it is forward*” (Product Owner, C3) and representing the team when interacting with the business “*if something is wrong I can tell the customer – “your feature has been dropped because of this”* (Product Owner, C3).

In C3, QA developed a greater sense of belonging and attachment to the team as a consequence of the SPM. This was only evident in C3 as this was the only team that had a QA resource that participated in the interviews (Table 4.6.4-1, p.115). It was not possible to determine if the QA resources in C1 (based in India) felt part of the team as they did not participate in the interviews conducted. However, they did participate in the SPM’s observed and were “*considered part of the team*” (Business Analyst, C1) by the team. In C3 the SPM provided the QA personnel with an opportunity to participate in and contribute to the project from the outset “*I prefer the sense that you have in being there from the beginning and being able to contribute... in the past where were simply just the gate keepers at the end ... I like that we [QA] have more of a contribution to the product*” (QA, C3), which was welcomed by QA.

There was some evidence in C3 that the team had developed a great sense of solidarity. While this could not be attributed to one single agile practice it was considered a consequence of using the various practices of Scrum. There was an instance in this team “*where a serious bug was found [in production]*” (Scrum Master, C3). Management were determined to find the person responsible for the mistake and pursued the matter until they did. The team “*was horrified about that*” (Developer 2, C3) and “*it did not feel right to name somebody*” (Scrum Master, C3),

“everybody was unhappy....there is now a much more sense of the team wins or the team loses kind of thing” (Developer 2, C3).

Members identify with the team

The SPM helped team members to identify with the team, which was discussed in section 5.1.1 on p.138.

Team interests are emphasised over self-interests

The SPM provided some evidence that team members placed emphasis on the interests of the team rather than their own personal interests. This was discussed previously in section 5.1.1 (p.138).

Members develop a strong team spirit

A strong team spirit existed in C2 and C3 with the team spirit described as “*pretty good*” (QA, C3) and “*positive*” (Developer 4, C2) with a “*good atmosphere*” present in the team (Developer 3, C3), which “*was good before, but I think it is better [since the introduction of Scrum]*” (Product Owner, C2). In C1 the views were mixed with the team spirit was described as “*pretty good*” (Developer 4, C1), “*ok*” (Developer 3 and Business Analyst, C1), “*Average, Medium*” (Project Manager, C1). The mixed views in this team were likely related to the recent redundancies in the organisation, although it was the Project Manager rather than the developers who had the most negative view of team spirit. He felt that the team “*don’t have a whole lot of loyalty to the company because the company hasn’t been loyal to them necessarily*” (Project Manager, C1). The strong team spirit in C2 and C3 could be attributed to the cooperation of all team members to establish the common goals for the sprint, their involvement in assigning tasks, estimating tasks, the regular involvement of the Product Owner in the SPM (C2, C3), and the regular involvement of QA in the SPM (C1, C3) as discussed in section 5.1.1 (p.138). This could also be due to the working environment or the culture in both of these organisations, which in C3 was described as “*cooperative*” (Developer 1, C3) and “*open and transparent*” (Developer 2, C3), although some of this could be attributed to the introduction of Scrum as “*it is a better way to work*” (Developer 2, C3), but also because “*we are all working on the one project really for 6 months*” (QA, C3). In C2 the working environment was

described as “*a cooperative culture*” (Developer 1, C2) where “we *solve problems together*” (Developer 2, C2) with employee satisfaction “*very high*” (Developer 2, C2) in this team, which is determined each year by an employee satisfaction survey. In C2 the strong team spirit could also be due to the length of time these team members are employed by the same organisation, with six of the nine individuals interviewed having over 10 years service, of which two had over 20 years service.

Characteristic of Clan Control: Members develop a strong sense of loyalty to the clan			
Factor	Case C1	Case C2	Case C3
Sense of attachment to the team	✓ SPM facilitates regular communication and interaction amongst team members	✓ SPM facilitates regular communication and interaction amongst team members	✓ SPM facilitates regular communication and interaction amongst team members
	No customer present in the team	✓ SPM facilitates regular participation of the Product Owner	✓ SPM facilitates regular participation of the Product Owner
	✓ SPM facilitates regular participation of QA in the team	No QA present in the team	✓ SPM facilitates regular participation of QA in the team
Sense of belonging and solidarity with the team	✓ SPM requires regular communication and interaction amongst team members	✓ SPM requires regular communication and interaction amongst team members	✓ SPM requires regular communication and interaction amongst team members
	No customer present in the team	✓ SPM facilitates regular participation of the Product Owner	✓ SPM facilitates regular participation of the Product Owner
	✓ SPM facilitates regular participation of QA in the team	No QA present in the team	✓ SPM facilitates regular participation of QA in the team
Sense of identity with the team	See Table 5.1.1-1		
Team interests are emphasised over self-interests	✓ SPM helps the team to work together to deliver on their agreed goals for the sprint e.g. assigning tasks to the most appropriate person	✓ SPM helps the team to work together to deliver on their agreed goals for the sprint e.g. sharing information and rotating tasks	✓ SPM helps the team to work together to deliver on their agreed goals for the sprint e.g. sharing information and rotating tasks
Strong team spirit	✓ SPM involves all team members in the establishment of common goals for the sprint	✓ SPM involves all team members in the establishment of common goals for the sprint	✓ SPM involves all team members in the establishment of common goals for the sprint
	X SPM permits the involvement of all team members in tasks e.g. task estimation, task assignment, which does not occur	✓ SPM permits the involvement of all team members in tasks e.g. task estimation, task assignment	✓ SPM permits the involvement of all team members in tasks e.g. task estimation, task assignment

Characteristic of Clan Control: Members develop a strong sense of loyalty to the clan			
Factor	Case C1	Case C2	Case C3
	No customer present in the team	✓ SPM promotes cooperation between the Product Owner and the remainder of the team e.g. agreement on prioritisation of tasks	✓ SPM promotes cooperation between the Product Owner and the remainder of the team e.g. agreement on prioritisation of tasks
	✓ SPM promotes cooperation between QA and the remainder of the team	No QA present in the team	✓ SPM promotes cooperation between QA and the remainder of the team

Table 5.1.3-1 SPM: Members develop a strong sense of loyalty to the clan

5.1.4 Members develop common interests and goals which evolve as tasks progress

As described in section 2.5.4, p.36, where clan control exists goals are typically unknown at the outset. Instead, clans develop common interests and goals, which evolve as tasks progress and as members become socialised into the clan and become familiar with the norms and values of the clan. In addition, where clan control exists in ISD teams they have autonomy over their goals. These two factors are examined in this section.

Common interests and goals develop, which evolve as tasks progress

In all three cases the SPM provided team members with an opportunity to develop common interests and goals, which evolved as the project progressed (Table 5.1.4-1). In ASD the goals naturally evolve from one sprint to the next depending on the requirements of the customer, changing priorities, and the resources available. This was previously discussed in section 5.1.1 (p.138).

Teams have autonomy over goals and how to achieve them

The team also had autonomy over their goals with responsibility allocated to the team to define and clarify the goals for each sprint and to determine how they were achieved, which was also discussed in section 5.1.1 (p.138).

Characteristic of Clan Control: Members develop common interests and goals which evolve as tasks progress			
Factor	Case C1	Case C2	Case C3
Common interests and goals develop, which evolve as tasks progress	✓ SPM recognises that short-term goals will evolve from one sprint to the next	✓ SPM recognises that short-term goals will evolve from one sprint to the next	✓ SPM recognises that short-term goals will evolve from one sprint to the next
	Customer not present in the team	✓ SPM requires participation of the Product Owner to ensure their goals and interests are aligned with the remainder of the team	✓ SPM requires participation of the Product Owner to ensure their goals and interests are aligned with the remainder of the team
Teams have autonomy over goals and how to achieve them	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of each task to one team members	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of tasks to two or more team members	✓ SPM provides teams with ownership over the short-term goals and how they are achieved e.g. assignment of each task to one team members
	✓ SPM provides individual team members autonomy over their personal goals e.g. team members can express a desire to work on particular tasks	✓ SPM provides individual team members autonomy over their personal goals e.g. team members share and rotate tasks	✓ SPM provides individual team members autonomy over their personal goals e.g. team members share and rotate tasks

Table 5.1.4-1 SPM: Members develop common interests and goals which evolve as tasks progress

5.1.5 Rituals and ceremonies identify and reinforce acceptable behaviours

Section 2.5.4 (p.38), described rituals and ceremonies as regular occurrences that “reinforce the implicit, but well understood codes of correct conduct” (Macintosh, p138). They are a source of information, communicate values and beliefs to members, require the recurring collective participation of members, require members to behave in a manner that is consistent with agreed-upon behaviours, and can be in the form of rules and regulations that are considered acceptable by the clan. This section discusses the ritual and ceremony that takes place during the SPM and also how the SPM identifies and reinforces acceptable behaviours.

Rituals and ceremonies

In ISD projects rituals and ceremonies occur in many ways such as meetings. In ASD the SPM is a meeting that occurred on a regular basis at the start of each sprint.

This meeting required team members to comply with certain rules and regulations, it reinforced certain behaviours to which team members agreed in order to be part of the ASD team, and it was a source of information for team members where the values and beliefs of the team could be communicated (Table 5.1.5-1). This section first describes the how each of the three teams conducted the ritual and ceremony of the SPM. While there were similarities across the three cases, each had their own nuances, which were a consequence of a number of factors, such as the environment they worked in, the distribution of the team, and the physical resources available to the team. This is followed by detail on the particular behaviours that developer and were considered acceptable for each team as a result of using the SPM. It is not intended to be an exhaustive list of behaviours, but to identify specific behaviours that resulted as a consequence of using the SPM. These behaviours were identified through observations and responses from interviewees and there were variations across each of the three teams as to what was considered acceptable behaviour. It was possible that the SPM encouraged other acceptable behaviours, which were not identified because interviewees did not consider them important, or it was something that occurred so regularly that they were not consciously aware of their behaviours or any subtleties of interactions between themselves and other team members. Additional observations of the teams over a sustained period of time could have identified additional behaviours. This section concludes with an examination of how the SPM reinforced those behaviours.

The SPM was conducted on a regular day, at the same time in the same location. The days and times of the SPM varied across the three cases to suit the requirements of the team and the environment within which the team operated. As all team members were expected to attend the SPM the times were selected to allow for maximum participation of team members, although in C1 team members did not always attend as discussed in section 5.1.1 (p.138). In Case C1 the SPM was conducted on a Friday afternoon in a large conference room after the sprint review and retrospective meeting (Obs C1). As this team was distributed across three locations the Project Manager reported that conducting the SPM and the sprint review and retrospective meeting consecutively avoided the necessity for an additional meeting. It also meant that team members based in Ireland and India could

proceed with their work on a Monday morning without the requirement to wait for the American team to commence their working day. Distributed team members from India and the United States participated via a conference call (Obs C1), which required team members in India to remain at work for several hours in order to attend the meeting, which occurred once every three weeks. In C2 the meeting was held at 8:30am on a Monday morning (Obs C2) in an area dedicated for the use of the Scrum team. Employees in this organisation had a flexible start time, but work normally commenced work at 8am with lunch time at 11:30am. The selection of 8:30am allowed time for all employees to be present at their place of employment, but it also allowed enough time for the meeting, so that it could ideally be completed by lunchtime. In C3, employees also had flexible working arrangements with employees permitted to commence work any time up until 10am. Therefore, a time of 10am was selected with the intention of completing the meeting by lunchtime at 12:30pm (Obs C3). Their meeting was held in the only large meeting room that existed within the organisation (Obs C3).

In all three cases **the duration of the meeting varied from one sprint to the next, but concluded within a defined time period.** The length of the meeting depended on the level of discussion and planning required by the team. In C1 the meeting observed lasted approximately 45 minutes, but it was reported by one developer that in the past these meetings could last hours even as long as a day and by another that “*the planning meeting tends to be quite short, 20-30 minutes because he [the project manager] has kind of already decided what we’re doing*” (Developer 2, C1). As this project has currently existed for two years meetings tend to be shorter now as the core development environment is in place “*there was a lot more to do back then because we were building a whole framework from scratch, now all your build stuff is there and your framework is there, you’re just adding bits on now*” (Developer 1, C1). In C2 the SPM observed had a duration of 3 hours, but it was reported by the Scrum Master the meetings often persist for a further hour. In C3 the meeting observed last approximately 90 minutes, but it was reported by the Scrum Master that meetings could have a duration of 2 hours.

Based on observations **each meeting followed its own individual ritual or process.** In C1 the Project Manager facilitated the SPM and was in control of the Product

Backlog (Obs C1). The Product Backlog was accessed via a web-based tool, which was projected onto a wall for the co-located team to view and was also visible to distributed team members via the organisational Intranet (Obs C1). The customer did not participate in the meeting and the distributed team members participated via a conference call (Obs C1). First, the priorities for the forthcoming sprint were discussed with the target number of story points for the sprint agreed by the team. Each user story was reviewed in turn with the “*In progress*” user stories reviewed first. User stories were updated with new tasks or sub-tasks added where required. In some instances the person assigned to the task was updated, but mostly the time estimate was updated to accurately reflect the expected completion date/time. Tasks were pre-assigned by the Project Manager before the meeting as he had an “*individual meeting with whoever is responsible for certain area and he’s kind of already decided what we’re doing*” (Developer 2, C1). Little discussion in relation to the estimates proposed for each task were observed as they were determined in advance of the meeting by the individual assigned to the task “*each developer does their own*” (Developer 1, C1) and were pre-populated on the web-based tool. The meeting concluded with a decision by the team on each user story/sub-task; for example, to move the task to the next sprint, or to ‘*Close*’ the task (Obs C1).

In C2 the meeting was facilitated by the Product Owner who controlled the Product Backlog, which was the focal point of the meeting with all discussion relating to specific items in the Product Backlog (Obs, C2). The Product Owner projected the Product Backlog, maintained in Excel, onto a wall for the entire team to view. First, the Product Backlog was reviewed with estimates for high-level tasks determined in story points. Tasks were added or deleted as each item was discussed in turn. Once the high-level tasks were agreed they were documented on postcards. A planning poker session followed to break the high-level tasks into lower level tasks and assign a number of hours to each story point. Planning poker cards were distributed to each team member and a discussion was held in relation to each item on the Product Backlog led by the person to whom it was assigned (Obs C2). This person outlined sub-tasks, which were written on the post-its and some discussion ensued, which on occasion required team members to use the whiteboard or flip chart available to demonstrate their point. Following the discussion each team member played a

planning poker card to determine an estimate and reach a consensus. In most cases it was observed that all developers played the same card (Obs C2). The agreed estimate was documented on the post-it that detailed the task. The post-its for all sub-tasks were attached to the postcard detailing the high-level task and the total time estimate was recorded on the Product Backlog (Obs C2). These post-its were later placed on the Scrum Board. Individual team members verified the number of hours available to them to work on the user stories and a sanity check was conducted by the Product Owner to ensure the team were optimised in terms of their workload (Obs C2). This sanity check was solely for the use of the team. The current value was “*0.5, which means that productivity has gone down by half*” (Product Owner, C2), but the Product Owner indicated to the team that he wanted this figure to be higher in the future.

In C3 the meeting was facilitated by the Scrum Master who was observed to take the lead and control the Product Backlog (Obs C3). The Product Owner in this case contributed when requested by the Scrum Master (Obs, C3). The Scrum Master first accessed the web-based tool that contained the electronic Product Backlog and burndown chart, which was projected onto the wall for the team to view (Obs C3). The Scrum Master displayed the burndown chart for the last sprint followed by the planning board and the task board (Obs C3). The task board consisted of three columns ‘*To-Do*’, ‘*In Progress*’ and ‘*Done*’. The team verbally agreed on their overall goal for the sprint, which was recorded on the web-based tool (Obs C3). This was followed by discussion between the Scrum Master and the Product Owner, who represented the customer, in relation to unknowns and prioritisation, and which functionality to include in the sprint. Even though the Product Owner claimed most requirements were equal prioritisation the Scrum Master requested that the Product Owner prioritise requirements. The Product Owner complied and electronically ordered the features with the high priority tasks listed first followed by lower priority tasks. Some new features were also added by the Product Owner. The team discussed which of the high priority tasks made sense to complete first and agreed on a suitable order. Once the order of tasks were agreed the team confirmed estimates for each task. Some tasks had indicative estimates recorded electronically in advance of the meeting, which were agreed verbally between the developers and QA. Tasks

that did not have time estimates assigned were determined by the developers who played planning poker. It was observed that the majority of the cards played for each task contained the same value, so agreement was quickly reached between the developers for each task. Where the values differed, discussion was held amongst the developers to agree a final estimate, which was then recorded on the web-based system. Following the planning poker session the Scrum Master detailed the complexity of tasks initially prioritised and the number of developers required to complete these tasks. QA informed the team of the requirements, limitations and current commitments of QA in relation to testing. A rough estimate was agreed on of 8-10 weeks development and 9-12 weeks QA work based on the availability of only one person for testing. A further discussion was held in relation to adding an additional resource to the project. It was agreed with the development manager to add an additional developer to the project for three sprints to allow for continued progress of the project during a period of annual leave for one developer.

Rituals and ceremonies identify acceptable behaviours

In each case the SPM provided the teams with autonomy to develop their own rules and regulations for the meeting. Each of the three teams had received formal training on the agile methodology and its practices and were aware of the format for the SPM as defined by their methodology. Yet, each team had developed their own rules, regulations, and behaviours that they considered acceptable. For example, all team members were expected to attend the SPM, but in C1 it was an acceptable behaviour not to attend the SPM, whereas in C2 and C3 it was unacceptable to be absent from the meeting. In addition, each SPM was chaired by the same individual, but this role was held by the Project Manager in C1, the Product Owner in C2, and the Scrum Master in C3. Similarly, in each of the three cases team members were expected to be punctual for the meeting and to contribute to the SPM in turn, to which they complied (Obs C1, C2, C3). Team members were also expected to contribute to discussions and provide help and feedback if required. The level of engagement in additional discussions varied across the three teams with individuals in C1 disengaging from the process when the discussion did not relate to their work (Obs C1) whereas team members regularly contributed in C2 and C3 (Obs C2, C3). It was possible that the level of engagement in C1 had diminished due to the length of time

that the teams were using the practice (two years), whereas teams in C2 and C3 the practice was only in use for a number of months and was still relatively new to the team (see Table 4.6.4-1 on p.115).

Rituals and ceremonies reinforce acceptable behaviours

In each of the three cases the SPM reinforced acceptable behaviour. The SPM required regular attendance at and participation in the SPM, which helped to reinforce the rules, regulations, and acceptable behaviours to team members, although as previously discussed members in C1 did not always attend. By attending the SPM team members quickly became familiar with the way the team operated, how team members interacted, what behaviours occurred, and what actions or behaviours were considered acceptable to the team. Attendance at the SPM was considered particularly beneficial for new team members as they helped “*to integrate you with so many people*” (Developer 3, C3) and new members can quickly become familiar with the norms of the team. Also, with the SPM it is “*pretty quick and easy to get up to speed in terms of what other people are working on*” (Project Manager, C1). It allowed team members to observe conformance with agreed behaviours and to sanction team members for non-conformance to behaviours. For example in C2 “*one developer did it [brought a laptop] a couple of times [to the SPM] because there was so much work that he had to do*” (Developer 4, C2). This individual was reprimanded for bringing a laptop to the SPM as it was not considered acceptable behaviour and they were asked to put it aside so that they could give their full attention to the meeting. In C3 there were “*certain norms of social behaviour that are unacceptable [to the team]*” (Developer 4, C3), of which this was one. No similar evidence found in C1, which may be due to the length of time that this team were using the SPM (Table 4.6.4-1 on p.115) as any unacceptable behaviour was likely eradicated some time ago.

Characteristic of Clan Control: Rituals and ceremonies identify and reinforce acceptable behaviour			
Factor	Case C1	Case C2	Case C3
Rituals and ceremonies	✓ SPM takes place at the same time in the same location at the start of each sprint	✓ SPM takes place at the same time in the same location at the start of each sprint	✓ SPM takes place at the same time in the same location at the start of each sprint
	✓ SPM concludes within a specified timeframe	✓ SPM concludes within a specified timeframe	✓ SPM concludes within a specified timeframe
	✓ Each SPM meeting follows a similar process	✓ Each SPM meeting follows a similar process	✓ Each SPM meeting follows a similar process
Rituals and ceremonies identify acceptable behaviour	✓ SPM provides the team with autonomy to develop rules and regulations for the meeting e.g. acceptable not to attend the meeting	✓ SPM provides the team with autonomy to develop rules and regulations for the meeting e.g. meeting is always chaired by the same individual	✓ SPM provides the team with autonomy to develop rules and regulations for the meeting e.g. expected to attend the meeting
	✓ SPM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance	✓ SPM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance	✓ SPM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance
Rituals and ceremonies reinforce acceptable behaviour	X SPM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour, but team members do not always attend	✓ SPM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour	✓ SPM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour,
	Not evident	✓ SPM allows the team to sanction team members for non-conformance to accepted behaviours	✓ SPM allows the team to sanction team members for non-conformance to accepted behaviours

Table 5.1.5-1 SPM: Rituals and ceremonies identify and reinforce acceptable behaviour

5.1.6 Members are influenced by shared norms, values and beliefs

As detailed in section 2.5.4 (p.39) members of a clan are influenced by shared norms, values and beliefs that exist within the clan with the clan correcting any deviant behaviour. Clan control is in operation when behaviour is influenced by shared norms, values and beliefs.

In all three cases the SPM resulted in the development of certain norms, values, and beliefs due to its ritual described in section 5.2.5 (p.194), which have somewhat influenced the behaviours of team members (Table 5.1.6-1). This section first

describes the how shared norms of the team influenced behaviour, which is followed by how shared values and beliefs influenced the behaviour of team members in each of the three cases. This was not intended to be an exhaustive list of shared norms, values and beliefs, but to identify specific examples that influenced behaviour in the three teams studied as a consequence of using the SPM. These behaviours were identified through observations and responses from interviewees. It was possible that other norms, values, and beliefs were present, which were not identified for the same reasons discussed in section 5.1.5 (on p.162).

Members are influenced by shared norms

The shared norms of the SPM influenced the behaviour of team members in different ways. There was an expectation in each of the three teams that everybody would voluntarily “*attend*” (Developer 3, C3) and “*contribute to*” (Developer 2, C2) the SPM to which team members in C2 and C3 complied. As discussed in section 5.1.1 (p.138) team members in C1 did not always conform to this norm, which did not appear to be a difficulty for the team (Obs C1). It was possible that individuals were reprimanded outside of the meeting, but this was not reported. It was more likely that this behaviour was considered acceptable to the team as team members were aware of the reason for their absence. Therefore, absence at a meeting was an acceptable behaviour, whereas it was considered unacceptable in C2 and C3 with team members conforming to the norm. In all three teams, members were actively encouraged to participate in the SPM and have an opinion with agreement reached by consensus rather than dictated by one individual (Obs C1, C2, C3), which was a “*change in behaviour*” (Scrum Master, C2) for team members. Where one person had an opinion “*others either criticise him or agree with him until we reach a consensus.*” (Developer 7, C2). “*It is more than one person’s opinion on what should be done ... everybody has their say into what they think that task is, how it should be split up, who should do it, or whatever else. It has just made the working environment a lot more open*” (Developer 2, C3). Team members felt that it was possible to “*give your opinion if you think that approach is not correct and that another approach would be better*” (Developer 3, C3), which was listened to by the team. Even though the collaborative process was reduced in C1 this did not mean that team members did not contribute to the meeting. Where specific tasks impacted on their work a

discussion was held between the relevant individuals to determine the most appropriate solution until an agreement was reached (Obs C1).

The SPM provided teams with the autonomy to allocate tasks as they wished, which encouraged the development of norms, which influenced their behaviour in how tasks were assigned. This was discussed in greater detail in section 5.1.1 (p.138) where in C1 the Project Manager assigned tasks to specific individuals, in C2 the team a norm emerged that tasks were assigned to two more individuals, and in C3 the team assigned tasks to specific individuals.

In C2 and C3 the Product Owner participated in the SPM, which influenced the interaction between the team and the Product Owner. In C3 the team used the SPM “*to negotiate and influence the Product Owner and to identify potential problems or things that are not that clear or can be misunderstood*” (Scrum Master, C3) and the Product Owner also used the meeting to clarify in person to the team exactly what was meant, or to “*try and put forward quite forcefully what I want.*” (Product Owner, C3). In C2 the Product Owner was treated differently by the team depending on the situation “*the team see me as the outsider when I put up goals for them [in the planning meeting]. But, when I deliver [to the customer] we are one unit delivering to someone else.*” (Product Owner, C2). The team in C2 on occasion needed to remind the Product Owner that “*the team cannot plan, if you have already planned it for us*” (Developer 1, C2) and realised over time that “*we need to say No [to the Product Owner], if we cannot do it*” (Developer 1, C2). In C1 the customer did not participate in the SPM although it was felt that “*the customer should be around when we are planning what to do in the next sprint*” (Technical Architect, C1), which resulted in the team “*assuming things*” (Developer 1, C1) and making decisions on behalf of the customer, which sometimes needed to be corrected at a later date.

The SPM also influenced the interaction between the team and the QA. In C3, QA were involved from the outset, which was a dramatic change for the way in which the team and QA worked together. Previously, “*QA would have limited their interaction altogether with them [development team] and now they would have no qualms. Everybody is talking to the developers and walking over to them and questioning them.*” (QA, C3). This was considered “*crucial*” (QA, C3) from a QA

perspective as the SPM provided an opportunity to communicate requirements, limitations, and commitments of QA to the team (Obs, C3), which resulted in a better understanding between the two groups. In C1, QA were also involved in the project from the outset and contributed to the SPM (Obs, C1), but as none of the QA personnel were interviewed in C1 it was not possible to present their views. However, the regular communication with the QA team in India during the SPM helped the Project Manager to keep track of their tasks and reinforce the goals of the team as “*we are talking every ten days and we actually started doing detailed sprint planning*” (Project Manager, C1). The team in C2 had no dedicated QA resources within the team, but they carried out their own quality checks “*we as a team are responsible for saying yes or no if the quality is okay or not*” (Developer 6, C2), with formal QA was conducted by a separate team within the organisation.

Members are influenced by shared values and beliefs

The SPM contributed to the development of shared values and beliefs amongst team members. Team members identified similar values and beliefs that were important to them such as “*learning*” (Developer 4, C2; Product Owner, C2), “*honesty*” (Developer 5, C2; Scrum Master, C2), “*cooperativeness*” (Developer 2, C1; Developer 6, Developer 7, C2; QA C3), “*communication*” (Developer 1, C3), and “*accountability*” (Product Owner, C2). This section examined the values that individuals within the teams identified as important to them and how the SPM reinforced each of these values and beliefs, which had some influence on the behaviour of team members. It was not intended to examine an exhaustive list of possible values and how the SPM influenced each. For example, learning and the sharing of knowledge and information was improved by the attendance at and contribution to the SPM as was communication as the SPM “*enhances the sharing of information*” (Developer 5, C2) and “*there is a lot of discussion about problems, how they should be solved, and how they relate to each other, so we learn a lot from that*” (Developer 4, C2). In C2 and C3 learning was further enhanced by the rotation of tasks amongst team members. If a team member was not familiar with any of the tasks on the Sprint Backlog they selected a task “*start working on it, and learn*” (Developer 3, C2). Team members were “*working on different things every day...people are getting full exposure to new things and to old things.. and are*

learning" (Developer 2, C3). Teams also worked together collaboratively to plan and estimate tasks in the SPM, which promoted cooperativeness and honesty amongst the team that the estimates "*are more accurate and more realistic*" (Developer 2, C3) and that the team could deliver on what was agreed. Accountability increased as team members felt accountable for their assigned tasks "*you're responsible for your part, your work, your task*" (*Technical Architect, C1*), "*if you take the task you feel responsible for it - to complete it*" (*Scrum Master, C3*) as well as feeling "*responsible for the performance of the team*" (Developer 7, C2).

Characteristic of Clan Control:			
Members are influenced by shared norms, values and beliefs			
Factor	Case C1	Case C2	Case C3
Shared norms influenced behaviour	X Team members are expected to regularly attend the SPM, but they may vary their levels of attendance	✓ Team members are expected to regularly attend the SPM and they always attend	✓ Team members are expected to regularly attend the SPM and they always attend
	X Team members are expected to contribute to the SPM in addition to their expected contribution, but levels of contribution can vary	✓ Team members are expected to contribute to the SPM in addition to their expected contribution	✓ Team members are expected to contribute to the SPM in addition to their expected contribution
	✓ Teams have autonomy to allocate tasks to team members, resulting in the assignment of each task to a single team member	✓ Teams have autonomy to allocate tasks to team members, resulting in the assignment of each task to a two or more team members	✓ Teams have autonomy to allocate tasks to team members, resulting in the assignment of each task to a single team member
	✓ Customer rarely participates in the SPM, which results in the team making assumptions in relation to their needs	✓ Product Owner always participates in the SPM	✓ Product Owner always participates in the SPM
	✓ QA always participate in the SPM, but they may limit their contribution, resulting in additional monitoring of their tasks by the Project Manager	No QA present in the team	✓ QA always participate in and contribute to the SPM, resulting in improved communication and interaction between QA and the remainder of the team
Shared values and beliefs influenced behaviour	X SPM encourages learning, which influences behaviour e.g. tasks are not rotated amongst team members	✓ SPM encourages learning, which influences behaviour e.g. tasks are rotated amongst team members	✓ SPM encourages learning, which influences behaviour e.g. tasks are rotated amongst team members

Characteristic of Clan Control: Members are influenced by shared norms, values and beliefs			
Factor	Case C1	Case C2	Case C3
	✓ SPM encourages cooperativeness, which influences behaviour e.g. team members participate in collective planning.	✓ SPM encourages cooperativeness, which influences behaviour e.g. team members participate in collective planning, assignment of tasks, estimating	✓ SPM encourages cooperativeness, which influences behaviour e.g. team members participate in collective planning, assignment of tasks, estimating
	✓ SPM encourages honesty, which influences behaviour e.g. promoted accuracy in estimates, realistic goal setting	✓ SPM encourages honesty, which influences behaviour e.g. promoted accuracy in estimates, realistic goal setting	✓ SPM encourages honesty, which influences behaviour e.g. promoted accuracy in estimates, realistic goal setting
	✓ SPM encourages communication, which influences behaviour e.g. team members vocalise opinions and views, communicate difficulties, provide feedback	✓ SPM encourages communication, which influences behaviour e.g. team members vocalise opinions and views, communicate difficulties, provide feedback	✓ SPM encourages communication, which influences behaviour e.g. team members vocalise opinions and views, communicate difficulties, provide feedback
	✓ SPM encourages accountability, which influences behaviour e.g. team members feel personally responsible to complete their assigned tasks	✓ SPM encourages accountability, which influences behaviour e.g. team members feel personally responsible to complete their assigned tasks	✓ SPM encourages accountability, which influences behaviour e.g. team members feel personally responsible to complete their assigned tasks

Table 5.1.6-1 SPM: Members are influenced by shared norms, values and beliefs

5.1.7 Members evaluate and reward or sanction each other

To reiterate what was stated in section 2.5.4 (p.40) clans evaluate, reward and sanction each other on their conformance to the agreed norms, goals, and values of the clan.

There was no evidence in any of the three cases to suggest that team members evaluated and rewarded each other for conformance to the team's norms, goals, and values (Table 5.1.7-1). The SPM provided a forum for teams to plan tasks for the upcoming sprint, prioritise tasks in conjunction with the customer, determine estimates for each task and assign them to individual team members for completion (section 3.3.2 on p.65), which were not conducive to evaluation or rewards.

The only evidence of team members evaluating and sanctioning each other for non-conformance to agreed norms, goals and values was only reported in C2 where a team member was reprimanded for bringing a laptop to the SPM “*One developer did it [brought a laptop] a couple of times because there was so much work that he had to do, and he was asked to put it away*” (Developer 4, C2), so that they could give their full attention to the meeting “*you should not be part time in the laptop and part time in the meeting*” (Developer 5, C2). Laptops were permitted at the SPM in C1 and several team members did bring their laptop to the meeting, which they accessed while attending the meeting (Obs C1). They were not permitted in C3 as this was considered generally unacceptable behaviour “*this is something people wouldn't do... there are certain norms of social behaviour that are acceptable*” (Developer 4, C3).

Characteristic of Clan Control: Members evaluate and reward or sanction each other on their conformance to the agreed norms, goals, and values			
Factor	Case C1	Case C2	Case C3
Evaluation of conformance to agreed norms, goals and values	No evidence that team members evaluated each others conformance to agreed norms, goals and values	✓ SPM allows team members to evaluate conformance of others to agreed norms, goals and values	No evidence that team members evaluated each others conformance to agreed norms, goals and values
Rewards for conformance to agreed norms, goals and values	Not applicable	Not applicable	Not applicable
Sanctions for non-conformance to agreed norms, goals and values	Not evident	✓SPM permits team members to sanction each other e.g. for non-conformance to the agreed norms, goals, and values	Not evident

Table 5.1.7-1 SPM: Members evaluate and reward or sanction each other

5.2 Daily Scrum Meeting

This section discusses each of the resultant characteristics of clan control as detailed in Figure 5.0-1 (p.137) and illustrates how the daily scrum meeting (DSM) accomplishes or fails to accomplish each characteristic in the three cases studied. It also leans on the factors of each resultant characteristic that were identified in the literature (section 2.5.4 on p.31) to assist with structuring the analysis and the presentation of the findings in each of the tables. To reiterate what was stated in

section 3.3.3 (p.66), the DSM is held each day of a sprint where team members detail what they have completed since the previous meeting, the tasks they will focus on for the coming day, and any impediments that are preventing them from completing their tasks.

5.2.1 Members exhibit a strong sense of identity with the clan

In section 2.5.4 (p.31) identification was defined as “*a psychological state wherein an individual perceives himself or herself to be part of a larger whole such as a work group, a team, or an organisation*” (Rousseau, 1998). Where individuals identify with a group, or team they have a feeling of attachment to the team, are loyal to the team, have similar goals, values and norms and are committed to following them, cooperate and work as a team, and self-interest decreases in favour of the teams. These are the five factors that are examined in this section.

The DSM contributed to the development of a sense of identity within the team in a number of ways (Table 5.2.1-1) based on the factors identified in the literature (Table 2.5.4-1, p.34). As stated in section 5.1.1 (p.138) team members recounted that they felt part of the team, although the Product Owners in both C2 and C3 stated that they did not always feel part of the team.

Members feel part of the team

Similar to the SPM the DSM helped team members to feel part of the team. The DSM was a daily touch point for each of the three teams where team members convened to briefly update the team on the progress and identify impediments preventing them from completing their daily tasks.

The DSM provided an opportunity for regular communication and interaction between team members, which was exhibited by attendance at and contribution to the daily meeting with each team member given an equal opportunity to speak. All team members engaged in discussion in addition to that which was expected; for example, providing feedback or advice to another team member. While team members in C1 did not always participate in the DSM this did not indicate that they did not feel part of the team. Similar to the SPM (p.138) their occasional absence was reported by team members present at the meeting as due to meeting conflicts

with other departments/personnel within the organisation, often based in the United States.

The DSM was important for building stronger relationships and communications between team members as the interaction on the DSM helped to “*understand the character of the different people on the team...you have a relationship with them as soon as possible and you are involved in what the team is doing*” (Developer 5, C3). It avoided people working on their own for long periods of time with little interaction with other team members. For example, it was not uncommon for developers in C2 to work “*for up to eight months basically doing the same very hard thing and then pop up ... like I made it, now it works..., but now they are really participating now and expressing opinions ...I perceive a very large change in the social interaction*” (Developer 1, C2).

In C1 and C2 the DSM immediately introduced new and distributed team members to each other and builds relationships. For example, “*just hearing the guys speak you can understand [better], like you might find a guy is very blunt and if you don't have those grouping conversations and you deal with him directly and you think it's to you he is being blunt, whereas you will hear him in stand-up, you would realise okay it's not me and then you feel more comfortable and that's just the way the person is, it doesn't bother you.*” (Developer 5, C1). In addition it helped a distributed team member in C1 “*feel part of the team and responsible for my tasks, due to the continuous [daily] communication between the team*” (Technical Architect, C1). It also helped to quickly integrate new personnel into the team even though “*they may not have much of a contribution*” at the outset (Developer 3, C1). The DSM provided the opportunity for both new and distributed team members to listen to discussions amongst other team members, which “*helped in terms of getting to know who is who and what is happening on the team*” (Developer 3, C1). As a consequence “*it's pretty quick – easy to get up to speed in terms of what other people are working on*” (Project Manager, C1). It also built an awareness of tasks assigned to each person “*straight away you know what someone new is up toespecially QA where they're not even co-located ... otherwise, you might never hear them and even for a few months you mightn't get an email out of them*” (Developer 1, C1). The Project Manager in C1 allowed a new co-located team member to “*settle in very easy and*

make it very clear in the first couple of weeks that you've probably nothing to say in stand-up, you've nothing to report, unless you want to say something... and I think people feel just more comfortable then." (Developer 5, C1). In C2 a new team member was quickly made feel part of the team in their first DSM where they discussed their assigned task when it was their turn to speak and answered the same three questions as other team members (Obs, C2). Consequently, he immediately had an opportunity to feel part of the team. C3 was a co-located team and did not contain any distributed team members, nor were any new team members introduced since the inception of Scrum.

Members are loyal to the team

The DSM provided evidence that team members were loyal to the team. This is discussed in greater detail in section 5.2.3, p.187.

Team members have similar goals, values, and norms and are committed to following them

Attendance at and contribution to the DSM in C2 and C3 helped team members to develop similar goals, values, and norms and illustrated their commitment to them. Team members interacted and communicated with each others, contributed to discussions, collaborated, and provided assistance to one other (Obs C2, C3). While similar interaction occurred in C1 team members did not always participate in the meeting (Obs C1), although the use of the DSM for two years likely meant that team members were already familiar with the norms and values of the team. Similar to the SPM new and distributed team members were expected to participate in the DSM from the outset in C1 and C2 (Obs, C2), which meant that they immediately had an opportunity to become familiar with the norms and values of the team. C3 was a co-located team and did not contain any distributed team members, nor were any new team members introduced since the inception of Scrum.

While the overall goal for the sprint was determined at the SPM team members conveyed their personal goals for the day at the DSM, which provided clarity on their daily goals (Obs C1, C2, C3). The DSM also provided visibility on daily progress and allowed the team to monitor the progress of others by making "the

process transparent ...it prevents you from straying too much from what we have planned" (Developer 1, C2). As progress was "visible...we can track progress and QA from the start" (Developer 2, C3) it helped team members "to keep track of where we are, how we are doing, and to help out when needed" (Developer 1, C2), to identify "areas that you know may potentially hold us up" (Developer 5, C1), or to determine "if we are late, or who may be late [delivering their functionality]" (Developer 1, C3). In C2 daily progress was updated on the Scrum Board (Figure 4.7.3-1, p.129) and the Burndown chart (Figure 4.7.3-2, p.130) (Obs C2), which were the focal point of the meeting as everyone can "see the [Scrum] board, the burndown chart... and we know what we have to finish" (Developer 2, C2). It became "very obvious ... what did I do yesterday" (Developer 2, C2) with the team able to quickly determine if the tasks for the sprint were progressing to the 'Done' column. There was a common interest in C2 to move a post-it to the 'Done' column on the Scrum Board (Obs C2), or to reach zero on the Burndown chart. In C1 and C3 the Scrum Board was not visible or used at the DSM as the Scrum Board in both of these teams was contained in a software tool used by the team. All of this ensured that team members remained focused on the task at hand in order to meet the deadline committed to during the SPM "*the [daily scrum] helps commitment ... it kind of enforces what you do in the team, for yourself as well as for others so, they keep you focused.*" (Business Analyst, C1). "*It makes you more focused and that should really end up with meeting the deadlines in a better way I think*" (Developer 1, C2). Consequently, there was little room for deviation from the task at hand, but kept team members committed to completing their daily tasks. In C1 the DSM also helped the Project Manager to monitor the work assigned to distributed team members as the QA team in India are "*fairly timid kind of guys, they don't really say much other than 'this is what I did' and 'this is what I'm going doing today*". (Developer 2, C1). The DSM gave you "*an insight into what they [distributed team members] are doing even if you're not in that area at least you know it's ticking along or it's on track as well*." (Developer 1, C1).

The DSM provided team members with ownership over their daily goals and helped to set and observe the completion of "*personal targets*" (Developer 3, C1), or to "*set more realistic goals*" (Developer 2, C2) to ensure that their tasks were completed as

promised. It was not possible for team members to become anonymous as tasks were transparent and visible to the team “*everything is open... we see the board this is what you have to do, and we see the information on the burndown chart ... everyone sees it*” (Developer 1, C2). It was welcomed by one developer that it was not possible for an individual to hide or to conceal a lack of effort “*I think it is better the more transparent and the more visible it is the better*” (Developer 3, C3). This suggests that this developer may have had previous experience on a project where it was difficult to determine the contribution of certain team members. The DSM also allowed team members to influence one another and the tasks they selected on a daily basis where there may be “*dependencies*” (Developer 1, C1) between development tasks. For example, where one developer may be “*working on a task that is blocking somebody else*” (Developer 4, C3) a developer may request that the developer focus on a different task for the day “*don't forget you have this to complete*” (Developer 2, C2) so that they can progress with theirs. In C1 and C3 these type of requests were sometimes made “*outside*” (Developer 2, C3) the DSM with team members not always waiting for the DSM to occur in order to make the request “*you would probably be more likely to contact that person directly by email, or in person*” (Developer 3, C1) as soon as it was identified that one developer needed another to complete their task in order to progress.

The DSM also facilitated the participation of the customer or Product Owner to ensure that their goals and the team goals were aligned for the duration of the sprint, although this did not occur in C1, as the customer rarely participated in the DSM. In C2 and C3 the Product Owner regularly participated in the DSM and was available to “*answer questions*” (Developer 2, C3), or the team could re-negotiate the goals for the sprint if it was realised that “*[a task] was not feasible to complete within the timeframe*” (Developer 3, C3). The opposite also occurred as illustrated in C2 where the Product Owner used the DSM to communicate “*problem reports [to team members that] we have to look at*” (Developer 3, C2). These additional tasks affected the amount of work that the team could achieve by the end of the sprint, which needed to be renegotiated by the Product Owner with the team.

In C1 the co-located team, based in Ireland, used the DSM to ensure that distributed team members were working towards the common goal, “*you would be checking*

with the DBA's [based in the USA] to ensure where they are, because we obviously have a higher dependency on them and it's nice just to clarify those" (Developer 5, C1). The Project Manager in C1 also used the DSM to keep track of the work assigned to the QA team based in India as he felt that "*they tend to give you a more positive picture of things, they don't want to tell you bad news and they don't want to tell you any difficulties*" (Project Manager, C1). This was not evident in C2 or C3 as there were no distributed team members in either of these cases.

Team members work and cooperate as a team

All three teams were very collaborative and worked and cooperated as a team. They were very receptive to helping one another in the DSM whenever assistance was required. For example, "*if you have a problem and you raise it in the [daily] scrum, everyone is very dynamic and thinking and it's so much more powerful than sitting there on your own [trying to resolve the problem]*" (Developer 1, C2), or "*if I have problems that we want to solve we can discuss it*" (Developer 2, C2), "*and then you get help*" (Developer 2, C1). If the team was not cooperative "*the team may fail as what one person could be doing will affect the other person*" (Developer 2, C3), but at the same time "*if you need help or don't understand something everyone is more than willing to help and that has improved now with Scrum*" (Developer 2, C3).

Team interests emphasised over self-interests

The willingness of individuals to assist each other when required illustrated that team members were willing to prioritise their own interests over those of the team. Assisting another team member with a task reduced the amount of time an individual had to complete their own tasks, but it was clear that the teams wanted to achieve their sprint goal and were willing to do whatever was necessary to accomplish that. There was no evidence to suggest that the DSM encouraged team members to put their own interests ahead of the team's interests.

Characteristic of Clan Control: Members exhibit a strong sense of identity with the clan			
Factor	Case C1	Case C2	Case C3
Members feel part of the team	✓ DSM is a daily touch point for the team	✓ DSM is a daily touch point for the team	✓ DSM is a daily touch point for the team
	X DSM provides an opportunity for team members to communicate and interact regularly, which does not always occur	✓ DSM provides an opportunity for team members to communicate and interact regularly	✓ DSM provides an opportunity for team members to communicate and interact regularly
	✓ DSM helps to build strong relationships between team members	✓ DSM helps to build strong relationships between team members	✓ DSM helps to build strong relationships between team members
	✓ DSM provides an opportunity for new and distributed team members to regularly participate in the team	✓ DSM provides an opportunity for new team members to regularly participate in the team	No new or distributed team members
Members are loyal to the team	See Table 5.2.3-1, p.191		
Team members have similar goals, values, and norms and are committed to following them	X DSM requires regular attendance at the meeting to become familiar with norms and values, which does not always occur	✓ S DSM PM requires regular attendance at the meeting to become familiar with norms and values	✓ DSM requires regular attendance at the meeting to become familiar with norms and values
	X DSM provides an opportunity for team members to contribute in addition to their expected contribution, which does not always occur	✓ DSM provides an opportunity for team members to contribute in addition to their expected contribution	✓ DSM provides an opportunity for team members to contribute in addition to their expected contribution
	✓ DSM provides an opportunity for new and distributed team members to participate in the DSM	✓ DSM provides an opportunity for new team members to participate in the DSM	No new or distributed team members present in the team
	✓ DSM provides clarity on daily goals	✓ DSM provides clarity on daily goals	✓ DSM provides clarity on daily goals
	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals e.g. burndown chart	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals
	✓ DSM provides team members with ownership over their daily goals and how they are achieved	✓ DSM provides team members with ownership over their daily goals and how they are achieved	✓ DSM provides team members with ownership over their daily goals and how they are achieved

Characteristic of Clan Control: Members exhibit a strong sense of identity with the clan			
Factor	Case C1	Case C2	Case C3
	X DSM permits participation of the customer to ensure their goals and the team goals remain aligned for the duration of the sprint, but the customer rarely participates	✓ DSM permits participation of the customer to ensure their goals and the team goals remain aligned for the duration of the sprint	✓ DSM permits participation of the customer to ensure their goals and the team goals remain aligned for the duration of the sprint
	✓ DSM ensures distributed team members had similar goals and interests to co-located team members	No distributed team members present in the team	No distributed team members present in the team
Team members work and cooperate as a team	✓ DSM highlights where team members have difficulties, which allows the team to provide assistance to each other	✓ DSM highlights where team members have difficulties, which allows the team to provide assistance to each other	✓ DSM highlights where team members have difficulties, which allows the team to provide assistance to each other
Team interests emphasised over self-interests	✓ DSM allows team members to identify where team members require help and they consequently provide the necessary assistance	✓ DSM allows team members to identify where team members require help and they consequently provide the necessary assistance	✓ DSM allows team members to identify where team members require help and they consequently provide the necessary assistance

Table 5.2.1-1 DSM: Members exhibit a strong sense of identity with the clan

5.2.2 Members exhibit a strong sense of commitment to the clan

In section 2.5.4 (p.34), team commitment was described as teams (clans) developing goals and values that members may accept, members exerting considerable effort on behalf of the team (clan), exhibiting a desire to remain a member of the team (clan), identifying with the team, and demonstrating loyalty to the team. These are the five factors that are examined in this section.

Team develop goals and values that members accept

The DSM contributed to the development of commitment to the team in a number of ways (Table 5.2.2-1) first of all by helping team members to develop similar goals and values as discussed in section 5.2.1 (p.176).

Members exert considerable effort on behalf of the team.

Teams also illustrated that they were willing to exert considerable effort on behalf of the team in the DSM. As discussed in section 5.1.2 (p.150) when the teams were under pressure to deliver their members worked overtime in order to meet agreed goals as they felt personally accountable for their tasks. A similar work ethic existed when team members were willing to work additional time to achieve their daily goals “*there is a commitment there to make that [task deadline], before “ah, its 5 o’clock – I am going home. I’ll leave it there”. Now, you might spend the extra 45 minutes to make sure you have it done, so it doesn’t creep into the next day*” (Developer 2, C3).

Evidence was also found in the three cases that team members were committed to expending additional effort to help each other, which was highlighted in the DSM “*if you have a problem that’s come up, you can mention it [in the daily scrum] and then you get help*” (Developer 2, C2). This provided an opportunity to “*get answers from everyone who has something to say about the issue.*” (Developer 3, C2), or was an opportunity for other team members to “*give their opinion*” (Developer 5, C1), “*provide feedback or suggestions how to do it differently*” (Scrum Master, C3), or to “*help you later [after the daily scrum]*” (Developer 2, C2), which in the view of this developer helped “*with the commitment of the team*” (Developer 5, C1). This contrasted with the manner in which individuals worked prior to the implementation of agile. Previously, team members felt “*more isolated and you didn’t really want to disturb anyone*” (Developer 2, C2) to request help. In C2 and C3 it was expected that “*if you cannot finish it [your assigned task] ...because of this problem... you have to say it in the daily scrum meeting*” (Developer 3, C3). This reflected the cooperative nature of team members, which may be due to the relationships that existed between team members, the length of time they were part of the same team, or the implementation of various agile practices, as they encouraged more communication and interaction.

In C1 the DSM was not always used to request help. Instead, team members asked for help immediately when it was required “*if you want to ask them you can just go over and tap them on the shoulder*” (Developer 5, C1). Team members did not feel

that they had to wait for a DSM to obtain an answer to their question or problem “*If I have a problem I'll go and ask a question ... somebody will always sit down with me*” (Business Analyst, C1). However, another developer felt that “*you wouldn't raise every little thing [in the daily scrum] you might say 'I'll get over that one myself' ... I suppose you don't want to sound like you've got problems on a daily basis.*” (Developer 1, C1). As this view was not reported by any other team members in C1 this opinion may be due to the level of experience of this developer, who was the “*first employee for that project*” (Developer 1, C1) and he may not wish to highlight to the team that he had a difficulty with a task.

The customer was expected to participate regularly in the team, as recommended by the methodology, see section 3.3.3 (p.66). In C1 the customer rarely participated in the meeting, which illustrated a lack of commitment by the customer to the team (Obs, C1) and unwillingness to expend effort on behalf of the team. Even though the team do not “*consider the customer part of the team*” (Project Manager, C1) their lack of participation made it difficult for the team to make decisions and on occasion “*we kind of assume things*” (Developer 1, C1), sometimes resulting in an incorrect decision, which needed to be corrected at a point in the future. One of the reasons for the lack of customer participation related to the software under development as it was a “*very technical back-end web services. There is no front-end. If you put the customer in front of us I doubt they would understand*” (Developer 1, C1). Also, as the project was so large there was no one single defined customer. Instead, there were “*a few different people in a few different areas...with no one person who understands it all*” (Developer 1, C1). In contrast in C2 and C3, the Product Owner regularly participated in the DSM and provided advice, direction and feedback to the team, which demonstrated their efforts and commitment to the team (Obs, C2, C3).

Characteristic of Clan Control: Members exhibit a strong sense of commitment to the clan			
Factor	Case C1	Case C2	Case C3
Team develop goals and values that members may accept	✓ DSM provides clarity on daily goals	✓ DSM provides clarity on daily goals	✓ DSM provides clarity on daily goals
	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals e.g. burndown chart	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals
	✓ DSM provides team members with ownership over their daily goals and how they are achieved	✓ DSM provides team members with ownership over their daily goals and how they are achieved	✓ DSM provides team members with ownership over their daily goals and how they are achieved
	Customer does not participate	✓ DSM permits participation of the customer to ensure their goals and the team goals remain aligned for the duration of the sprint	✓ DSM permits participation of the customer to ensure their goals and the team goals remain aligned for the duration of the sprint
	✓ DSM ensures distributed team members had similar goals and interests to co-located team members	No distributed team members present in the team	No distributed team members present in the team
Members exert considerable effort on behalf of the team (clan)	✓ DSM ensures team members feel personally accountable for their tasks	✓ DSM ensures team members feel personally accountable for their tasks	✓ DSM ensures team members feel personally accountable for their tasks
	✓ Team members willingly help each other when assistance is requested, but assistance is rarely requested in the DSM	✓ Team members willingly help each other when assistance is requested in the DSM	✓ Team members willingly help each other when assistance is requested in the DSM
	Customer does not participate	✓ DSM provides an opportunity for the Product Owner to participate in the meeting e.g. provide advice and feedback	✓ DSM provides an opportunity for the Product Owner to participate in the meeting e.g. provide advice and feedback
Exhibit a desire to remain a member of the team (clan)	Not evident	Not evident	Not evident
Members identify with the team	See Table 5.2.1-1, p.183		
Members are loyal to the team	See Table 5.2.3-1, p.191		

Table 5.2.2-1 DSM: Members exhibit a strong sense of commitment to the clan

Members exhibit a desire to remain part of the team.

Similar to the SPM there was no evidence in any of the three cases to suggest that the DSM had a specific influence over individuals and their desire to remain part of the team. As discussed in section 5.1.2 (p.150) team members indicated that they “*like the way this team work*” (Developer 5, C3), they “*have a very good way as team*” (Scrum Master, C2) since the introduction of their selected agile methodology. Individuals had no desire to work on a team that “*don’t follow agile*” (QA, C3) with a “*preference for agile to the traditional approach*” (Developer 3, C1), but this was due to the agile methodology as a whole rather than to specific agile practices.

Members identify with the team

The DSM contributed to the development of a sense of identification with the team in a number of ways, which was discussed in section 5.2.1 on p.176.

Members are loyal to the team

The DSM also contributed to the development of a sense of loyalty to the team, which are discussed in greater detail in section 5.3.3 on p.218.

5.2.3 Members develop a strong sense of loyalty to the clan

In section 2.5.4 (p.35), loyalty to an organisation was defined as “*a feeling of affection for and attachment to an organisation*” (Buchanan, 1974), which can also be applied to a team or a group. Individuals who are loyal to a clan or a team develop a feeling of affection for and attachment to the team, identify with the team, develop a sense of belonging and solidarity with the team, team interests are emphasised over self-interests, and they develop a strong team spirit. These are the five factors that are examined in this section (see Table 5.2.3-1).

To a certain degree the DSM contributed to a strong sense of loyalty to the team. The DSM increased visibility and transparency on tasks across all three teams. Team members conveyed that they developed a greater sense loyalty to the team as a result of the visibility and transparency, as they had a desire to complete all tasks that they had undertaken to achieve. They wanted “*to do themselves justice*” (Developer 4, C1) with one developer feeling that he was “*letting the team down*” (Developer 2, C2) if

he did not complete his tasks on time. The DSM “*gives a pressure, an automatic pressure on what you are doing, because we show what we are doing*” (Developer 1, C3) where team members felt that they were “*responsible for my tasks not someone else*” (Technical Architect, C1). It was particularly evident in C2 as this team displayed their Scrum Board on the office wall for all to see with focus placed on the Scrum Board during the DSM (Obs C2). “*Everything is open – we see the [Scrum] board every day – this is what you have to do – everyone sees it.*” (Developer 2, C2), with the Scrum Master of the view that the increased transparency has encouraged greater loyalty in the team as he felt “*they are better [more loyal] now than before*” (Scrum Master, C2). Both C1 and C3 used electronic scrum boards, which were not accessed or visible during the DSM.

The DSM promoted daily interaction and open communication between team members across the three teams, “*if you have a good team, open, good team that works well together, it encourages loyalty*” (Developer 2, C2). According to the Project Manager in C1 “*anything that facilitates greater conversation*” encourages greater loyalty in his view. The DSM provided a forum for individuals to easily request assistance “*where people identify that others are struggling in a certain area, they will either give their opinions or they will assist in other ways to follow up action outside the stand-up*” (Developer 5, C1) or, “*if somebody has a problem and somebody else says ‘I can help you’ you have a naturally formed bond if they have trouble the next time you’re more likely to help.*” (Developer 1, C2), all of which helped to develop loyalty amongst individual team members. Another team member believed that “*it’s not my assignment. It’s our assignment, so to speak*” and if help was required to complete a task it as requested - “*if I need help, I’ll get help directly from someone*” (Developer 2, C2). Only the Scrum Master in C3 reported that increased interaction and communication helped to increase loyalty by “*influencing how you feel towards the other members of the groupin a positive way....previously, you would not talk that much to other people*”. Other team members in C3 reported that they did feel loyal to the team, but this loyalty was not caused by the DSM. For example, “*loyalty is an attribute I have always had*” (QA, C3) with another of the view that the ‘experts’ in the team have demonstrated more loyalty. “*I feel a little bit more loyalty from some of the experts so to speak*”

(Developer 6, C2) as team members now had the opportunity to ask experts for help/feedback in the DSM, and it was expected that help would be provided. Help from experts was more difficult to obtain previously because “*he wasn’t involved in that so it was hard to get feedback*” (Developer 6, C2).

Members develop a sense of attachment and a sense of belonging to the team

The DSM helped to develop a sense of attachment and a sense of belonging to the team through daily communication and interaction between all team members, including distributed team members. The daily interaction between team members was described as encouraging “*togetherness in a way, ‘we’re all in it together’ type of thing.*” (Developer 2, C2). Each individual was required to contribute to the DSM by answering the three questions typically asked in a DSM (section 3.3.3 on p.66): what was completed since the last meeting, what they intended to do today and what, if any impediments they were experiencing that may prevent them from achieving their daily goal (Obs, C1, C2, C3)

The DSM facilitated customer participation in the meeting, but in C1 the customer did not “*have visibility into our daily stand-ups*” (Business Analyst, C1). Therefore, there was no obligation for the customer to participate and there appeared to be little loyalty between the customer and the team with the customer not considered part of the team. A team member from Ireland “*was on-site [the previous year] for 6 months in Boston with the customer...and I think that definitely did help*” (Business Analyst, C1) as this person had greater clarity on the needs of the customer and reported their requirements to the team during the DSM. This contrasted with the regular participation of the Product Owner in the DSM in both C2 and C3. In C3, the Product Owner attended the DSM “*4 out of 5 times I would say....if I don’t get pulled into something else*” (Product Owner, C3). This demonstrated their feeling of attachment and sense of belonging to the team as they made themselves available to the team to provide feedback if required. In C2 a different approach was employed and the attendance of the Product Owner at the DSM varied intentionally “*as we are seeing different dynamics in the team when I am there and when I am not there. Or, when I say something, or when I am silent. So, we have tried to find a balance of how often should I be there*” (Product Owner, C2). Even when the Product Owner

did not attend the meeting they were available and accessible to the team to provide feedback if required.

Similar to the SPM distributed team members, QA, and new team members participated in the DSM. This increased communication and interaction and helped them to feel part of the team. Like the SPM, QA in C3 felt that they “*have more of a contribution to the product*” (QA, C3) as a consequence of participating in the DSM. However, the QA team in C1, who were distributed, participated in a separate DSM solely with the Project Manager, due to time zone differences. It is unknown if participation in this DSM helped them to feel part of the team as these team members did not participate in the study. The remaining distributed team member in C1, who had an opportunity to participate in the DSM with the core development team welcomed the daily communication as a means to keep informed of the progress of the team as it helped him to “*feel part of the team*” (Technical Architect, C1). C2 contained a new team member who participated in the DSM from the outset (Obs C2). This immediate participation in the DSM should contribute to the development of a sense of belonging to the team.

Members identify with the team

The DSM contributed to team members developing a sense of identification with the team, which was discussed in greater detail in section 5.2.1 on p.176.

Team interests are emphasised over self-interests

Similarly, as discussed in section 5.2.1 (p.176) the DSM provided some evidence that team members were willing to prioritise their own interests over those of the team.

Members develop a strong team spirit

As described in section 5.1.3 (p.156) a strong team spirit existed in C2 and C3 with mixed views of the strength of the team spirit present in C1. The DSM contributed to this to a certain extent through the cooperative nature of team members and their willingness to provide assistance to each other if required in order for the team to achieve the sprint goal. This was discussed in more detail in section 5.2.2 (p.183).

Characteristic of Clan Control: Members develop a strong sense of loyalty to the clan			
Factor	Case C1	Case C2	Case C3
Strong sense of loyalty	✓ DSM increases visibility and transparency within the team	✓ DSM increases visibility and transparency within the team	✓ DSM increases visibility and transparency within the team
	✓ DSM involves daily team interaction and communication	✓ DSM involves daily team interaction and communication	✓ DSM involves daily team interaction and communication
Sense of attachment to the team	✓ DSM requires daily communication and interaction amongst team members	✓ DSM requires daily communication and interaction amongst team members	✓ DSM requires daily communication and interaction amongst team members
	No customer present in the team	✓ DSM facilitates daily participation of the Product Owner	✓ DSM facilitates daily participation of the Product Owner
	✓ DSM facilitates daily participation of QA in the team, but at a time that is different to the DSM for the core team	QA not present in the team	✓ DSM facilitates daily participation of QA in the team
	✓ DSM facilitates daily participation of distributed team members	✓ DSM facilitates regular participation of new team members	No distributed or new team members present in the team
Sense of belonging and solidarity with the team	✓ DSM requires daily communication and interaction amongst team members	✓ DSM requires daily communication and interaction amongst team members	✓ DSM requires daily communication and interaction amongst team members
	No customer present in the team	✓ DSM facilitates daily participation of the Product Owner	✓ DSM facilitates daily participation of the Product Owner
	✓ DSM facilitates daily participation of QA in the team	QA not present in the team	✓ DSM facilitates daily participation of QA in the team
	✓ DSM facilitates daily participation of distributed team members	✓ DSM facilitates regular participation of new team members	No distributed or new team members present in the team
Sense of identity with the team	See Table 5.1.1-1, p.150		
Team interests are emphasised over self-interests	✓ DSM allows team members to identify where team members require help and they consequently provide the necessary assistance	✓ DSM allows team members to identify where team members require help and they consequently provide the necessary assistance	✓ DSM allows team members to identify where team members require help and they consequently provide the necessary assistance
Strong team spirit	✓ Team members willingly help each other when assistance is requested in the DSM, but assistance is rarely requested	✓ Team members willingly help each other when assistance is requested in the DSM	✓ Team members willingly help each other when assistance is requested in the DSM

Table 5.2.3-1 DSM: Members develop a strong sense of loyalty to the clan

5.2.4 Members develop common interests and goals which evolve as tasks progress

In section 2.5.4 (p.36) detail was provided on the development of common interests and goals in a clan. Where clan control exists goals are typically unknown at the outset. Instead, clans develop common interests and goals, which evolve as tasks progress and as members become socialised into the clan and become familiar with the norms and values of the clan. In addition, where clan control exists in ISD teams they have autonomy over their goals. These two factors are examined in this section.

Common interests and goals develop, which evolve as tasks progress

In all three cases the DSM provided team members with an opportunity to develop common interests and goals, which evolved as the project progressed (Table 5.2.4-1), which were discussed in section 5.2.1 (p.176).

Teams have autonomy over goals and how to achieve them

Each of the three teams had autonomy over their goals within the sprint and could change the goals where it became obvious to the team that all tasks would not be completed within the timeframe allocated. This occurred in all three cases for example; “*if you don’t get it done you move it to the next sprint*” (Developer 1, C1), “*if a certain feature is going to make something late, you have the option of dropping it or making something late*” (Product Owner, C3), or “*we have had a couple of sprints where the work that we estimated was much more than could be done.*” (Developer 7, C2). This resulted in the reallocation of tasks to the subsequent sprint if it was determined that they would not be achieved.

C2 had a particular challenge as this team had multiple goals with team members required to work on three projects at the same time. This made it “*very hard for the individuals, to really, really focus as a team, because you are supposed to maybe support other things as well right now*” (Scrum Master, C2). From the developers perspective “*it’s kind of messy where you have a couple, or more than one goal for a sprint maybe, you have 3 or 4 and also other stuff that pops in pulling on people’s time*” (Scrum Master, C2). Working on multiple projects had occasionally resulted in the team in C2 “*re-planning things mid-sprint because a piece of work comes in*

from the side" (Developer 1, C2), which was not ideal from the teams perspective as it affected their ability to deliver the functionality agreed for the sprint. According to the Product Owner out of the 17 completed sprints replanning had only occurred "*once or twice, but we should have done it at least two or three more times*" (Product Owner, C2). This also occurred in C3 if "*there was something new that happened that we didn't plan*" (Developer 1, C3). There was no evidence in C1 that the team replanned or changed goals mid-sprint. This may be because when the team were under pressure to deliver functionality within a tight time-frame they were overburdened with functionality and did not have the time to re-plan or could not consider re-planning as there was already an excessive amount of work to complete. Now that the pressure has alleviated somewhat on the team, their work has become slightly segmented in the current release as the team resolve errors with the previous release reported by the customer. The nature of this work has not required any re-planning mid-sprint.

Characteristic of Clan Control: Members develop common interests and goals which evolve as tasks progress			
Factor	Case C1	Case C2	Case C3
Common interests and goals develop, which evolve as tasks progress	✓ DSM provides clarity on daily goals	✓ DSM provides clarity on daily goals	✓ DSM provides clarity on daily goals
	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals e.g. burndown chart	✓ DSM provides visibility on daily progress and allows the team to monitor progress of members towards daily goals
	✓ DSM provides team members with ownership over their daily goals and how they are achieved	✓ DSM provides team members with ownership over their daily goals and how they are achieved	✓ DSM provides team members with ownership over their daily goals and how they are achieved
	Customer not present in the team	✓ DSM permits participation of the customer to ensure their goals and the team goals remain aligned for the duration of the sprint	✓ DSM permits participation of the customer to ensure their goals and the team goals remain aligned for the duration of the sprint
	✓ DSM ensures distributed team members had similar goals and interests to co-located team members	No distributed team members present in the team	No distributed team members present in the team

Characteristic of Clan Control: Members develop common interests and goals which evolve as tasks progress			
Factor	Case C1	Case C2	Case C3
Teams have autonomy over goals and how to achieve them	✓ DSM provides team members with ownership over their daily goals and how they are achieved	✓ DSM provides team members with ownership over their daily goals and how they are achieved	✓ DSM provides team members with ownership over their daily goals and how they are achieved
	✓ DSM permits the movement of tasks to subsequent sprints if they cannot be achieved within the current sprint	✓ DSM permits the movement of tasks to subsequent sprints if they cannot be achieved within the current sprint	✓ DSM permits the movement of tasks to subsequent sprints if they cannot be achieved within the current sprint
	Not evident	✓ DSM allows goals to change mid-sprint if required	✓ DSM allows goals to change mid-sprint if required

Table 5.2.4-1 DSM: Members develop common interests and goals which evolve as tasks progress

5.2.5 Rituals and ceremonies identify and reinforce acceptable behaviours

Section 2.5.4 (p.38) described rituals and ceremonies as regular occurrences that “*reinforce the implicit, but well understood codes of correct conduct*” (Macintosh, p138). They are a source of information, communicate values and beliefs to members, require the recurring collective participation of members, require members to behave in a manner that is consistent with agreed-upon behaviours, and can be in the form of rules and regulations that are considered acceptable by the clan. This section discusses the ritual and ceremony that takes place during the DSM and also how the DSM identifies and reinforces acceptable behaviours.

Rituals and ceremonies

In ISD projects rituals and ceremonies occur in many ways such as meetings. In ASD the DSM was such a meeting that occurred daily during each sprint. This meeting required team members to comply with certain rules and regulations, it reinforced certain behaviours to which team members agreed in order to be part of the ASD team, and it was a source of information for team members where the values and beliefs of the team could be communicated (Table 5.2.5-1). This section first describes the how each of the three teams conducted the ritual and ceremony of the DSM. While there were similarities across the three cases, each had their own

nuances, which were a consequence of a number of factors, such as the environment they worked in, the distribution of the team, and the physical resources available to the team. This is then followed by detail on the particular behaviours that were considered acceptable for each team as a result of using the DSM. It is not intended to be an exhaustive list of behaviours, but to identify specific behaviours that have resulted as a consequence of using the DSM. These behaviours were identified through observations and responses from interviewees. There were variations across each of the three teams as to what was considered acceptable behaviour. It was possible that the DSM encouraged other acceptable behaviours, which were not identified because interviewees did not consider them important, or it was something that occurred so regularly that they were not consciously aware of their behaviours or any subtleties of interactions between themselves and other team members. Additional observations of the teams over a sustained period of time could identify additional behaviours. This section concludes with an examination of how the DSM reinforced those behaviours.

The DSM was **conducted at the same time each day during the sprint in the same location**. The time selected for the DSM varied across the three cases, but was consistent within each team. In C1 two separate DSM's were held to cater for the distributed nature of the team. The Project Manager conducted one DSM via a conference call at 10am Irish time with the Indian team. The second was conducted at 4pm Irish time to facilitate the American team, also via a conference call, which was held in a conference room with the remainder of the team based in Ireland. The Project Manager communicated the content of the DSM with the Indian team to the remainder of the team at the 4pm meeting. In C2 the DSM was conducted at 8:45am in a dedicated open area located beside the team, which contained the Scrum Board (Obs C2). As this organisation had flexible working hours this time was selected to allow team members flexibility in their start time, but also to allow team members to have their usual morning coffee break at 9am on conclusion of the meeting. In C3 the DSM was conducted at 10am in a small meeting (Obs C3). This time was selected to also cater for flexible working arrangements within this organisation as all employees were expected to be in the office by 10am at the latest.

In all three cases the **duration of the meeting varied from one DSM to the next** depending on the level of discussion that occurred. In C1, the DSM between the Irish and American team typically had duration of 10-15 minutes (Obs C1), but it was reported that meetings have lasted up to 30 minutes. This occurred when all team members were present and a discussion ensued that then turned into a project update covering items that were only of interest to specific individuals. The duration of the meetings observed in C2 were between 15 and 30 minutes (Obs C2), but the Scrum Master reported that the DSM had on occasion lasted an hour or more depending on the level of discussion that takes place “*the daily scrum goes up and down and my observation is that it depends on the difficulty and uncertainty of the tasks in the sprint.*” (Scrum Master, C2). This was supported by a developer who felt that the team tended “*to go into detailed discussions a lot in the daily scrum*” (Developer 6, C2) with “*the longest [daily scrum] lasting more than 1 hour*” (Developer 3, C2). In C3 the DSM had a typical duration of 15 minutes (Obs C3) and it was reported by the development manager that DSM’s do not tend to exceed this time and are often shorter.

Based on observations **each meeting followed its own individual ritual** or process. In C1 the meeting commenced once the distributed team member was contacted and available on a conference call. Team members stood in a circle around the meeting room (Obs, C1) as this was part of the ritual for the meeting as indicated by the agile practice 3.3.3 (p.66). The facilitator of the meeting rotated from one meeting to the next (Obs C1). All team members were expected to attend the meeting, which did not occur on occasion as discussed in section 5.2.1 (p.176). All team members present contributed to the meeting where each team member in turn briefly updated the team on their progress (Obs, C1). Team members asked for clarification on tasks, which occasionally resulted in some discussion. However, the team minimised discussion and where it appeared that a long discussion may take place the individuals concerned were requested by other team members to hold additional discussions after the meeting (Obs, C1). If this required the input of the distributed team member they were requested to remain on the conference call to participate.

In C2 the team commenced the meeting without prompting from the Scrum Master. The team stood in a semi-circle facing the Scrum Board with the Product Owner and

the Scrum Master standing slightly back from the team in more of an “*observer role*” (Product Owner, C2). This was intentional on their part as in their view the team were in charge of the meeting and their role was not to tell the team how to do the work, but to “*help the team get better*” (Scrum Master, C2). All team members were expected to attend the meeting, which they did in the meetings observed (Obs C2). The facilitator role rotated in each meeting observed (Obs C2). Team members conducted two rounds to answer the three questions typically asked at a DSM as prescribed by the Scrum methodology (p.66).

- What did you do yesterday?
- What do you plan to do today?
- Are there any impediments?

In the first round each developer spoke in turn detailing what they did the previous day. The first person to speak was the person closest to the Scrum Board. The speaker held a marker with the marker passed in turn from one developer to the next until it reached the last (Obs, C2). As each developer spoke they either moved a post-it from one column to another on the Scrum Board (not necessarily the next column), or updated the information on a post-it (for example, a time estimate). If progress was not made, or if the status of the task had not changed no post-it was moved or updated. Discussion took place amongst the developers on several tasks with the Scrum Master and Product Owner contributing or intervening when required. They reported after the meeting that their limited contribution was intentional on their part as where possible they wanted the team to resolve the discussion themselves. Where a discussion became heated or continued for too long the Scrum Master intervened to break-up the discussion (Obs C2). The Scrum Master spoke last and updated the columns on the right-hand side of the Scrum Board with the time remaining, followed by an update of the Burndown chart displayed on the wall beside the Scrum Board (Obs, C2). In the second round all developers spoke in turn again and stated what they planned to do today and detailed any impediments they faced. Once each developer had spoken and any issues had been resolved the Scrum Master concluded the meeting (Obs, C2).

In C3 all team members stood in a circle around the meeting room as this was part of the ritual for the meeting as indicated by the agile practice 3.3.3 (p.66). All team members were expected to attend the meeting with the exception of the Product Owner in C3. However, the Product Owner in C3 regularly attended the meetings “*I am at it 4 out of 5 times*” (Product Owner, C3), but did not contribute unless requested by the team. The meetings were facilitated by the Scrum Master (Obs C3), although he was trying to change this practice “*I am thinking that I will ask somebody every week to be responsible that the meeting starts on time*” (Scrum Master, C3). In one of the meetings observed the Scrum Master asked another team member to initiate the DSM the following week (Obs C3). All team members present contributed to the meeting (Obs C3). The Scrum Master chaired the meeting and asked the person to his right to start by answering two of the three questions typically asked at a DSM as prescribed by the Scrum methodology (page 66):

- What did you do yesterday?
- What do you plan to do today?

Team members spoke in turn and briefly stated what work they did the previous day and what they planned to do today (Obs, C3). Some discussion took place between team members on one or two occasions. Where it was recognised that further discussion was required team members agreed to hold a separate discussion after the meeting for the relevant personnel (Obs, C3). Once each team member had spoken the Scrum Master reported on his progress. This was then followed by a general question posed by the Scrum Master to the team to determine whether there were any issues blocking anyone (Obs, C3). Team members were not required to answer it in turn and they responded where they had a difficulty (Obs, C3).

Rituals and ceremonies identify acceptable behaviours

In each case the DSM identified acceptable behaviours for the team, which varied across the three teams. Each of the three teams had received formal training on the agile methodology and its practices and were aware of the format for the DSM as defined by their methodology. Yet, each team developed their own rules, regulations, and behaviours that they considered acceptable. For example, all team members

were expected to attend the DSM, but in C1 it was an acceptable behaviour not to attend the DSM, whereas in C2 and C3 it was unacceptable to be absent from the meeting. Likewise, the facilitator of the DSM was rotated in C1 and C2 (Obs C1, C2), but was held by the Scrum Master in C3 (Obs C3). Team members in all three cases were expected to be punctual for the DSM and to contribute to the DSM in turn, to which they complied (Obs C1, C2, C3). Team members were also expected to contribute to discussions and provide help and feedback if required, which they also did (Obs C1, C2, C3).

A particular behaviour emerged in C2 as a consequence of implementing the DSM, which had become acceptable to the team. At the DSM team members “*don’t go and update the figure [on the post-it]*” (Product Owner, C2) where the time required for a task took longer than initially estimated. Neither did the team question the lack of action “*no-one is asking that question... there is no team accountability*” (Product Owner, C2). However, this was contradicted to some extent by one developer who indicated that “*if someone is doing something that is out of context and you have eight hours on the little yellow slip and it doesn’t ever decrease, than people sort of question what are you doing*” (Developer 1, C2). However, it is possible that this only occurred where a task remained incomplete for several days. For the most part failure to update estimates appeared to have become an acceptable behaviour within the team, which was also observed in a DSM (Obs, C2) and had yet to be corrected by the team. This resulted in a false portrayal of progress, but the team did not seem to consider this a problem even though the Product Owner did not agree with this behaviour. This also occurred in C3 where “*sometimes the guys forget to update estimates or to set a task in progress, although this practiced isn’t accepted or encouraged*” (Scrum Master, C3) with the one developer indicating that “*we are actually quite poor at updating that [estimates], so, yeah, I suppose it is accepted then by the team that you don’t update the estimate.*” (Developer 2, C3). This may have occurred in C1, but it was not reported or observed.

A developer in C3 reported another behaviour that had become acceptable within their team for a while before it was corrected. This developer was one of the more experienced individuals on the project and he had started to make throwaway statements without considering their implications on the remainder of the team. For

example, “*I wasn’t bothered doing that test properly...I just stubbed it and went on past it. I didn’t know how to do it.*” (Developer 2, C3). Other team members had interpreted these statements as “*that’s fine, you will get away with that*” (Developer 2, C3) and had changed their behaviour accordingly until this developer realised what was happening. Consequently, he had to change his behaviour to ensure that this did not happen in the future and that other team members also did not continue with this behaviour.

Rituals and ceremonies reinforce acceptable behaviours

In each of the three cases the DSM reinforced acceptable behaviours. The DSM required daily attendance at and participation of all team members, which helped to reinforce rules, regulations, and acceptable behaviours to team members. This did not always occur in C1 as previously reported. The DSM was particularly useful for new team members as they quickly became familiar with the way the team operated, how team members interacted, what behaviours occurred, and what actions or behaviours were considered acceptable to the team (Obs C2). The DSM also allowed team members to observe conformance with agreed behaviours and to sanction team members for non-conformance to behaviours. This was observed in C2 where a team member was sanctioned when they spoke out of turn (Obs C2). Team members in C3 were sanctioned for late arrival, which “*was clamped down on like a tonne of bricks*” (QA, C3). This resulted in the Scrum Master “*keeping a record of attendance*” (QA, C3) for a while in order to correct this behaviour. There were no reports of team members in C1 sanctioning each other for non-conformance to any behaviours, which may be due to the length of time that this team are using the DSM (Table 4.6.4-1) with any unacceptable behaviours likely eradicated some time ago.

Characteristic of Clan Control: Rituals and ceremonies identify and reinforce acceptable behaviour			
Factor	Case C1	Case C2	Case C3
Rituals and Ceremonies	✓ DSM takes place at the same time in the same location at the start of each sprint	✓ DSM takes place at the same time in the same location at the start of each sprint	✓ DSM takes place at the same time in the same location at the start of each sprint
	✓ DSM concludes within a specified timeframe	✓ DSM concludes within a specified timeframe	✓ DSM concludes within a specified timeframe
	✓ Each DSM meeting follows a similar process	✓ Each DSM meeting follows a similar process	✓ Each DSM meeting follows a similar process
Rituals and ceremonies identify acceptable behaviour	✓ DSM provides the team with autonomy to develop rules and regulations for the meeting e.g. acceptable not to attend the meeting	✓ DSM provides the team with autonomy to develop rules and regulations for the meeting e.g. acceptable not to update estimates	✓ DSM provides the team with autonomy to develop rules and regulations for the meeting e.g. meeting is always chaired by the same individual
	✓ DSM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance	✓ DSM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance, order of speaking failure to update estimates	✓ DSM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance, failure to update estimates
Rituals and ceremonies reinforce acceptable behaviour	X DSM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour, but team members do not always attend	✓ DSM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour e.g. correction when team members speak out of turn, failure to correct individuals when an estimate is not updated	✓ DSM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour
	Not evident	✓ DSM allows the team to sanction team members for non-conformance to accepted behaviours	✓ DSM allows the team to sanction team members for non-conformance to accepted behaviours

Table 5.2.5-1 DSM: Rituals and ceremonies identify and reinforce acceptable behaviour

5.2.6 Members are influenced by shared norms, values and beliefs

As detailed in section 2.5.4 (p.39) members of a clan are influenced by shared norms, values and beliefs that exist within the clan with the clan correcting any deviant behaviour. Clan control is in operation when behaviour is influenced by shared norms, values and beliefs.

In all three cases the DSM resulted in the development of certain norms, values, and beliefs due to its ritual described in section 5.2.5 (p.194), which have somewhat influenced the behaviours of team members (Table 5.2.6-1). This section first describes the how shared norms of the team influenced behaviour, which is followed by how shared values and beliefs influenced the behaviour of team members in each of the three cases. This was not intended to be an exhaustive list of shared norms, values and beliefs, but to identify specific examples that influenced behaviour in the three teams studied as a consequence of using the DSM. These behaviours were identified through observations and responses from interviewees. It was possible that other norms, values, and beliefs were present, which were not identified for the same reasons discussed in section 5.2.5 (on p.194).

Members are influenced by shared norms

The shared norms of the DSM influenced the behaviour of team members in different ways. There was an expectation in each of the three teams that everybody would “*attend*” (Developer 3, C3) the DSM to which team members in C2 and C3 conformed. As discussed in section 5.2.1 (p.176) team members in C1 did not always conform to this norm. It was also expected that “*we are on time for the daily scrum meetings*” (Developer 3, C3). In C2 and C3 team members were occasionally late for meetings, which was “*not considered acceptable*” (Developer 5, C2) with team members verbally sanctioned and “*told to be on time*” (Developer 2, C3). This was not reported in C1, which may be due to the distributed nature of the team as distributed team members participated via a conference call, which was pre-arranged for a specific time. In addition, the selection of the time for the DSM resulted in a change in behaviour for team members in C2 with team members changing their habits in relation to the time they commenced work. “*Before Scrum they arrived to work about half past nine or something like that.*” (Developer 5, C2). Team members now commenced work by 8:45am at the latest. A similar change in behaviour was not observed in C1 or C3 as the DSM was held in the afternoon in C1 and in C3 the DSM was held at 10am, the latest time by which employees were expected to be in work.

Team members were expected to “contribute to” (Developer 2, C2) the DSM. However, in C1 the norm with the distributed team members based in India was to refrain from reporting “bad news or any difficulties and that tends to be a bit of problem” (Project Manager, C1), whereas the norm with the American team members was “to be a bit more positive about things and kind of brush things under the carpet when there might be a potential issue” (Project Manager, C1). This required the Project Manager to adjust his behaviour and to carefully monitor progress on their activities, which was assisted by the regular communication at the DSM. In addition, the DSM changed the behaviour of individuals with those that were previously “perceived as quiet are talking more now than before” (Developer 6, C2), everyone now “speaks up a lot more than ever before” (Developer 2, C3), and “everyone has an opinion” (Product Owner, C2). Team members were “participating and expressing opinions” (Developer 1, C2) and where they thought “that an approach was not correct and that another approach would be better” (Developer 3, C3) this was communicated.

The norm in the DSM was to report daily progress, which influenced the behaviour of some team members. In C3 some team members “watched the progress on certain tasks, especially if they were important” (Developer 2, C3) whereas in C2 team members were not always “good at keeping each other on the map [keeping on top of one another’s work] so to speak” (Developer 1, C2). However, individuals in this team were expected to work on multiple projects in a single sprint where the amount of time allocated to each project was defined in the SPM. Any additional unplanned work had become very obvious since “we got the Scrum Board” (Developer 2, C2). If it was discovered that team members were “working on the side” (Developer 2, C2) and “doing things that they should not, the team react” (Developer 1, C2) by asking the person to “skip that, do not do that, forget about it” (Developer 1, C2). The increased transparency and visibility of the DSM meant that “team members cannot stray off and just spend two or three days in investigations of whatever, you have to try and stay focussed on your task in hand.” (Developer 1, C2). They also found that they no longer had time to “delve down into a problem... as it would impact the [delivery for the] whole sprint” (Developer 1, C2).

A norm had developed in C2 and C3 to request and provide support to each other when needed “*you can ask more easily for it [help]... it is the norm now to ask for help*” (Developer 1, C3), but “*you need to raise it [the problem] as soon as you possibly can*” (Developer 1, C2) and “*the entire team is there to support*” (Developer 2, C2). There was an expectation that help would be requested if required “*it's completely acceptable to ask questions whenever you need help*” (Developer 2, C2), but if a team member did not request help “*it would be seen in a negative light*” (Developer 2, C3). The behaviour in C1 was different where the DSM was in use for two years. Sometimes team members “*forget the three questions they should be asking themselves and they forget that this is your chance to raise an issue and see if anybody else can help you.*” (Developer 1, C1). One team member does “*not necessarily raise them [problems] in the daily stand up... sometimes they just think they'll fix it themselves eventually*” (Developer 1, C1) as this person did not want to “*be annoying other people...as they will have to stop what they're doing*” (Developer 1, C1). This person was very experienced on the project as he was the “*the first employee for the project*” (Developer 1, C1), so it was possible that he preferred to resolve any problems himself. While he claimed he was comfortable raising an issue in the DSM he also did “*not want to sound like he had problems on a daily basis*” (Developer 1, C1). Other team members indicated that they will ask for help when required, although they do not necessarily wait for the DSM to request help “*you can just go over and tap them on the shoulder*” (Developer 5, C1) as team members “*know who to contact for a particular area*” (Developer 2, C1).

Members are influenced by shared values and beliefs

Similar to the SPM as discussed in section 5.1.6 (p.169) the DSM contributed to the development of shared values and beliefs amongst team members in all three teams i.e. learning, honesty, cooperativeness, communication, and accountability. The DSM reinforced each of these values and beliefs. For example, communication and learning was improved by the attendance at and contribution to the DSM. Learning was further enhanced by listening to the difficulties that team members raised and the potential solutions identified. When required teams worked together collaboratively to provide feedback and help to each other to deliver on their assigned tasks as it was “*the norm now to ask for help [if needed]*” (Developer 1, C3), which promoted

cooperativeness amongst the teams. The DSM also kept “*people honest*” (Project Manager, C1), particularly with distributed team members as team members were held accountable for their work and were easily “*exposed*” (Product Owner, C2) if there was a failure to demonstrate progress. Or, if progress is not reported “*it raises questions*” (Project Manager, C1) as to what the individual has done, or whether they have difficulties that have not been communicated. In C2, the Scrum Board and Burndown chart were visible to all, which detailed who was assigned to each task and an estimate for the task, and the progress of tasks (Obs C2), which “*indicated how it [the project] goes*” (Product Owner, C2), which reinforced accountability in this team. It was easy to determine “*what people are working on and why they are late*” (Developer 1, C3) and it was also easy to quickly identify where progress was poor, although this was not always evident in C2 where as discussed team members did not always update an estimate if a task was taking longer than expected.

Characteristic of Clan Control: Members are influenced by the shared norms, values and beliefs			
Factor	Case C1	Case C2	Case C3
Shared norms influenced behaviour	X Team members are expected to regularly attend the DSM, but they may vary their levels of attendance	✓ Team members are expected to regularly attend the DSM and they always attend	✓ Team members are expected to regularly attend the DSM and they always attend
	✓ Team members are expected to contribute to the DSM in addition to their expected contribution, which has resulted in increased communication by all team members	✓ Team members are expected to contribute to the DSM in addition to their expected contribution, which has resulted in increased communication by all team members	✓ Team members are expected to contribute to the DSM in addition to their expected contribution, which has resulted in increased communication by all team members
	✓ Distributed teams do not always communicate accurate progress, resulting in additional monitoring by the Project Manager	No distributed team members present in the team	No distributed team members present in the team
	No evidence to suggest that this occurred	✓ DSM increases transparency and visibility and ensures team members stay focused on the task at hand and do not deviate	✓ DSM increases transparency and visibility, resulting in team members monitoring each others work more closely

Characteristic of Clan Control: Members are influenced by the shared norms, values and beliefs			
Factor	Case C1	Case C2	Case C3
	X DSM encourages collaboration where team members immediately assist each other, but help is not always requested when required	✓ DSM encourages collaboration where team members immediately request assistance and help each other	✓ DSM encourages collaboration where team members immediately request assistance and help each other
Shared values and beliefs influenced behaviour	✓ DSM encourages learning, which influences behaviour e.g. problems are shared and assistance is provided	✓ DSM encourages learning, which influences behaviour e.g. problems are shared and assistance is provided	✓ DSM encourages learning, which influences behaviour e.g. problems are shared and assistance is provided
	✓ DSM encourages cooperativeness, which influences behaviour e.g. team members provide assistance to other team members on tasks, if required	✓ DSM encourages cooperativeness, which influences behaviour e.g. team members provide assistance to other team members on tasks, if required	✓ DSM encourages cooperativeness, which influences behaviour e.g. team members provide assistance to other team members on tasks, if required
	✓ DSM encourages honesty, which influences behaviour e.g. team members must be honest when reporting progress	✓ DSM encourages honesty, which influences behaviour e.g. team members must be honest when reporting progress	✓ DSM encourages honesty, which influences behaviour e.g. team members must be honest when reporting progress
	✓ DSM encourages communication, which influences behaviour e.g. team members vocalise opinions and views, communicate difficulties, provide feedback	✓ DSM encourages communication, which influences behaviour e.g. team members vocalise opinions and views, communicate difficulties, provide feedback	✓ DSM encourages communication, which influences behaviour e.g. team members vocalise opinions and views, communicate difficulties, provide feedback
	✓ DSM encourages accountability, which influences behaviour e.g. failure to report progress results in questions asked	✓ DSM encourages accountability, which influences behaviour e.g. visibility of tasks on scrum board encourages team members to complete their tasks as expected	✓ DSM encourages accountability, which influences behaviour e.g. ability to identify who is responsible for a particular task

Table 5.2.6-1 DSM: Members are influenced by shared norms, values and beliefs

5.2.7 Members evaluate and reward or sanction each other

To reiterate what was stated in section 2.5.4 (p.40) clans evaluate, reward and sanction each other on their conformance to the agreed norms, goals, and values of the clan. The DSM provided team members with an opportunity to evaluate and

reward or sanction each others conformance to the agreed norms, goals, and values of the team, although evidence was limited (Table 5.2.7-1).

Members evaluate each other on their conformance with the team's norms, goals, and values

There was some evidence of evaluation against conformance to the team's norms, goals, and values. As discussed in section 5.2.2 (p.183) the DSM provided a greater awareness and visibility of the amount of effort spent on each task and who had completed a task as this information was recorded in the Product Backlog and was visible on the Scrum Board (Obs C1, C2, C3). It was easy for team members to evaluate each others progress against agreed goals, and quickly identify a lack of effort or a lack of progress. For example, it was used to evaluate when "*people are struggling in a certain area*" (Developer 5, C1) so that assistance could be provided.

Members reward each other for conformance to the team's norms, goals, and values

There was some evidence in C2 of rewarding other team members for achieving their daily goals. It was observed at a DSM in C2 that when a developer moved a post-it from the 'In Progress' Column to the 'Done' on the Scrum Board that the team provided praise and recognition in the form of a cheer and a clap (Obs C2). The Scrum Master stated that is was a common occurrence when a team member completed a task. Similar evidence was not found in C1 or C3, which may be because the Scrum Board was not utilised or visible at the DSM (Obs C2, Obs C3) as the Scrum Board in both of these teams was contained in a software tool used by the team. Neither was verbal praise provided in either C1 or C3 when a team member reported the successful completion of a task (Obs C1, C3). This may also be because the meetings in both C1 and C3 tended to be consistently short (approx 10-15 minutes in duration) with team members quickly selecting their next task when one was completed. In C1 it may also be due to the length of time that the practice was in use, resulting in a routine practice where team members quickly progressed to the next task.

Developers across all three teams felt some level of personal satisfaction when finishing their assigned tasks. They enjoyed reporting the successful completion of goals to the team, which they found personally rewarding. For example, “*I get appreciation from the team when I say what I’ve done*” (Developer 6, C2) or “*accomplishing what you set out to accomplish is a sort of a reward in itself... if you have done a good job then there is a certain amount of pride in that*” (Developer 2, C3), or when “*we see our efforts going into a production system... someone will ultimately end up using that is going to make their lives or jobs easier and so you do get an awful lot of satisfaction from that*” (Developer 3, C1).

Members sanction each other for non-conformance to the team’s norms, goals, and values

There was little appetite in any of the three teams studied to sanction each other in any way with one team member indicating that “*there’s no conflict really, or there’s no criticism*” (Developer 2, C2). The feeling of another developer was that “*if there is criticism it is correct criticism*” (Developer 7, C2), which can be “*recognised in the tone sometimes, but it’s not an open criticism, I think it’s rather people still tend to use arguments that have to do with the subject rather than the person, so that’s rather good I would say*” (Developer 6, C2). While team members in C2 did reproach each other on occasion this did not appear to be a problem within the team, as it seemed that any criticism should be taken on board and was for the betterment of the team “*I personally appreciate that because it’s better to know it now.*” (Developer 6, C2). Likewise, in C3 there was little evidence of team members sanctioning each other, but evidence was provided that team members were sanctioned when they did not conform to an agreed norm. Individuals were “*told to be on time*” (Developer 2, C3) if they were regularly late for a DSM. This was also evident in C2 where “*it was not really accepted that you are late*” (Developer 5, C2), but “*it’s not a big deal really*” (Developer 1, C2) as “*we try to be in time as good as we can.*” (Developer 1, C2). Team members in C2 were also sanctioned for speaking out of turn in the DSM (Obs C2). There was no evidence observed or reported in C1 of team members sanctioning each other in any way at the DSM, which may be due to the length of time that this team are using the DSM (Table 4.6.4-1) with any unacceptable behaviours likely eradicated some time ago.

The lack of evidence of team members sanctioning each other may also be due to cooperative nature of the teams, which was reported by 20 of the 25 individuals interviewed. None of the other 5 team members indicated that team members were uncooperative. In C2 every team member indicated that the team were very cooperative, which may be due to the length of service these individuals have to the team and the organisation and their level of familiarity with each other (Table 4.6.4-1, p.115). Even though team members in C1 and C3 had reasonable length of service to their respective organisations, only one member of team C3 (Product Owner) did not indicate whether the team was cooperative or not. Even though he interacted with the team on a daily basis he did not interact with the team to the same extent as they would with each other, which was a consequence of the role that this individual held. Only half of the team members interviewed in C1 commented on the cooperative nature of the team. While this did not mean that other team members considered team members uncooperative it may be due to the manner in which tasks were allocated in this team. Each team member tended to have competencies in a particular area and completed tasks that fitted within this remit, so there was a reduced requirement for team members to cooperate with each other.

Characteristic of Clan Control: Members evaluate and reward or sanction each other on their conformance to the agreed norms, goals, and values			
Factor	Case C1	Case C2	Case C3
Evaluation of conformance to agreed norms, goals and values	✓ DSM allows team members to evaluate each other's progress against daily goals	✓ DSM allows team members to evaluate each other's progress against daily goals	✓ DSM allows team members to evaluate each other's progress against daily goals
	✓ DSM allows team members to evaluate each other for conformance to the agreed norms, values and goals of the team	✓ DSM allows team members to evaluate each other for conformance to the agreed norms, values and goals of the team	✓ DSM allows team members to evaluate each other for conformance to the agreed norms, values and goals of the team
Reward for conformance to agreed norms, goals and values	Not evident	✓ Team members reward each other on successful completion of goals	Not evident
Sanction for non-conformance to agreed norms, goals and values	Not evident	✓ DSM allows team members to sanction each other for non-conformance to agreed norms, goals, and values	✓ DSM allows team members to sanction each other for non-conformance to agreed norms, goals, and values

Table 5.2.7-1 DSM: Members evaluate and reward or sanction each other

5.3 Sprint Review and Retrospective

This section discusses each of the resultant characteristics of clan control as detailed in Figure 5.0-1 (p.137) and illustrates how the sprint review and retrospective meeting (SRRM) accomplishes or fails to accomplish each characteristic in the three cases studied. It also leans on the factors of each resultant characteristic that were identified in the literature (section 2.5.4 on p.31) to assist with structuring the analysis and the presentation of the findings in each of the tables. To reiterate what was stated in section 3.3.4 (p.68) a sprint review and retrospective is a meeting that is held at the end of each sprint where team members demonstrate the software completed in the sprint to the customer and management. They further highlight what worked well in the sprint and what did not, are encouraged to provide constructive feedback to each other, and identify actions that need to be addressed for future sprints.

5.3.1 Members exhibit a strong sense of identity with the clan

In section 2.5.4 (p.31) identification was defined as “*a psychological state wherein an individual perceives himself or herself to be part of a larger whole such as a work group, a team, or an organisation*” (Rousseau, 1998). Where individuals identify with a group, or team they have a feeling of attachment to the team, are loyal to the team, have similar goals, norms and values and are committed to following them, cooperate and work as a team, and self-interest decreases in favour of the teams. These are the five factors that are examined in this section.

Members feel part of the team

The SRRM contributed to team members exhibiting a strong sense of identity (Table 5.3.1-1) based on the factors identified in the literature. Similar to the SPM and the DSM the SRRM was a regular touch point for all team members where they convened at the end of each sprint to review and reflect on the sprint and work completed, which helped team members to feel part of the team. The SRRM required regular communication and interaction amongst team members, which was demonstrated by team members attending the meeting, participating in the meeting when expected, and providing input in addition to their expected contribution,

although in C1 team members did not always attend (Obs C1). However, their lack of attendance did not mean that individuals did not feel part of the team as all team members in C1 reported that they felt part of the team.

In C1 the regular communication and interaction between distributed team members and the remainder of the team at the SRRM helped to integrate distributed team members into the team “*you get to hear their [distributed team member] comments.....when the Technical Architect came on board they said in one of the retrospectives what they thought could be done better so you get feedback from them straight away, it's not 3 weeks.*” (Developer 1, C1). It also helped the co-located members to identify with the distributed team members and consider them part of the team. In C2 the only new team member since the implementation of Scrum had not yet participated in an SRRM, so it was not possible to determine if the SRRM helped them to feel part of the team. In C3 no new team members had joined since the introduction of Scrum, so it was not possible to infer if this meeting helped new team members to feel part of their team.

Members are loyal to the team

There was evidence to indicate that the SRRM contributed to the development of a sense of loyalty to the team. See section 5.3.3 on p.218 for further discussion.

Team members have similar goals, values, and norms, and are committed to following them

At the SRRM teams developed similar goals, values, and norms and were committed to following them. For example, in C2 and C3 team members demonstrated working software to the customer, ensured that their goals were met as agreed at the SPM, reviewed and reflected on whether the goal of the sprint was achieved with the primary focus of determining areas of strength and areas of weakness, and to identify actions on how the team as a whole can improve as described in section 3.3.4 (p.68).

The SRRM illustrated the teams’ commitment to the expected norms and goals of the team as there was “*an expectation that all team members would attend the sprint retrospective*” (Developer 3, C3) with team members in C2 and C3 regularly attending the meeting. However, participation of team members in C1 was

sometimes irregular for the same reasons discussed in section 5.1.1 (p.138) and section 5.2.1 (p.176), which had become the norm in this team. In C1 the participation of distributed team members in the SRRM suggested that they developed a familiarity with the norms and values of the team and of the meeting as each team member was given an equal opportunity to speak (Obs C1, C3). In C1 the contribution of team members was limited with team members briefly reflecting on the work they had completed during the sprint (Obs C1). There was no additional discussion amongst team members in the SRRM observed in C1. The SRRM was not observed in C2, but it was reported that team members contributed to the meeting in addition to their required contribution. In C3 team members engaged in additional discussion in addition to that which was expected; for example, discussion in relation to the improvement of quality (Obs C3).

The SRRM also provided an opportunity for team members to identify areas for improvement in future sprints, for the customer to participate in the review part of the SRRM, and for QA, where present to participate in and contribute to the SRRM. Each of these is discussed further in the next four paragraphs to avoid repetition.

Team members work and cooperate as a team

Each of the similar goals, norms and values developed encouraged the team to work cooperatively with each other, with QA, and with the customer, where present. Software was not demonstrated to the customer in C1 as the customer rarely attended the SRRM and also because of the nature of the software developed. The software developed in this team had no user-interface, so it was not possible for a customer, if they were present, to interact with the software. Lack of participation by the customer in the SRRM in C1 resulted in a lack of cooperation between the customer and the remainder of the team. In C2 and C3 the team collectively demonstrated the software produced in the sprint to the Product Owner for review and feedback on goals achieved (Obs C3, Reported by Product Owner in C2). Team members responded to questions asked by the Product Owner and management (Obs C3), which helped to increase cooperation between the Product Owner and remainder of the team.

In the SRRM observed in C1 each team member, including QA, briefly reflected on the work they had personally completed in the sprint (Obs C1). There was little interaction or cooperation observed between team members. Team members in C1 indicated that the meeting was “*not used to its full potential*” (Developer 1, C1) and had “*little value*” (Project Manager, C1) with one developer considering them “*pointless because the thing is built*” (Developer 5, C1). This may be due to the nature of the work recently as the work in the current series of sprints related to maintenance of the previous release rather than core software development, which will commence again in the near future. However, the Project Manager is attempting to change the format of the SRRM “*we've started collecting metrics... so we can do a compare and foster more conversation out of that, so I think that's where they're going to start getting a bit more value out of it.*” (Project Manager, C1).

This contrasted with C2 where the SRRM was considered “*inspiring and you can really hear what everyone thinks*” (Developer 7, C2), “*a very good thing, to think back and analyse how we worked*” (Developer 1, C2) with the team using the retrospective to “*talk about difficulties*” (Developer 4, C2), which they tried to address and where possible “*we take that into account in the following sprint planning*” (Developer 5, C2). Likewise, in C3 the SRRM was used to review and reflect on the previous sprint and identify difficulties (Obs, C3). However, the views in C3 were conflicting where some team members felt that “*we don't use it [SRRM] fully*” (Scrum Master, C3), whereas another developer felt “*they were very beneficial*” (Developer 3, C3), and another felt they were only beneficial “*after a few sprints, not just after one sprint ... as most of the time there is nothing really particular that you can observe in just two weeks, but, after you know, a few sprints, you can have a big picture*” (Developer 5, C3).

As prescribed by the methodology in section 3.3.4 (p.68) the SRRM was utilised by a team to reflect on the sprint completed, identify areas of strength and weakness, identify lessons learned and detail actions for improvement in subsequent sprint. In C1 minimal discussion or interaction was observed at the meeting in relation to identifying strengths, weaknesses, or lessons learned, with no actions identified (Obs C1). No cooperation between team members was observed with team members briefly answering in turn to the questions asked by the Project Manager with the

meeting concluding within 15 minutes (Obs C1). No issues were identified in the retrospectives observed in C1, although it was reported by the Project Manager in C1 subsequent to one observation that these meetings can sometimes become quite animated with copious discussion. This contrasted with observations in C3 where team members, including QA, engaged in the meeting, participated in the discussion, and collaborated on decisions made, identified areas for improvement and agreed on actions taken (Obs C3). While the SRRM was used to “*set objectives that we have to get; for example 80% of the coverage in unit test*” (Developer 3, C3, Obs C3) it was also actively used by QA to determine “*how we can improve things [from a QA perspective]*” (QA, C3). Others in C3 felt that “*we have discussions with each other which are very interesting, but then there are no actions behind this*” (Developer 1, C3), or “*there are no very specific actions as a result*” (Scrum Master, C3). The SRRM was not observed in C2.

Team interests emphasised over self-interests

There was no evidence in the SRRM in any of the three teams studied that this meeting was used by team members to put their own interests ahead of those of the team. The purpose of the meeting was not to complete work, or to assign work, but to review the software delivered and the way the team worked in the sprint; for example, in relation to how they interacted, the technical aspect of the product, and any difficulties encountered. The intention was to remove any impediments with the aim of improving the future productivity of the team. Therefore, the focus was naturally on the team and the best interests of the team as opposed to the interests of individuals. It is not surprising that there was no evidence to suggest that the SRRM encouraged individuals to put their own interests first. However, it may also be due to the limited use of the practice, which was particularly evident in C1.

Characteristic of Clan Control: Members exhibit a strong sense of identity with the clan			
Factor	Case C1	Case C2	Case C3
Members feel part of the team	✓ SRRM is a frequent touch point for the team	✓ SRRM is a frequent touch point for the team	✓ SRRM is a frequent touch point for the team
	X SRRM provides an opportunity for regular communication and interaction amongst team members, which does not always occur	✓ SRRM provides an opportunity for regular communication and interaction amongst team members	✓ S SRRM PM provides an opportunity for regular communication and interaction amongst team members
	✓ SRRM provides an opportunity for continuous team effort and participation from distributed team members	New team member had not yet participated in an SRRM	No new or distributed team members present in the team
Members are loyal to the team	See Table 5.3.3-1, p.221		
Team members have similar goals, values, and norms , and are committed to following them	X SRRM requires regular attendance at the meeting, which does not always occur	✓ SRRM requires regular attendance at the meeting	✓ SRRM requires regular attendance at the meeting
	X SRRM provides an opportunity for team members to contribute in addition to their expected contribution, which does not always occur	✓ SRRM provides an opportunity for team members to contribute in addition to their expected contribution	✓ SRRM provides an opportunity for team members to contribute in addition to their expected contribution
	No customer present in the team	✓ SRRM requires participation of the Product Owner to ensure their goals as agreed in the SPM were met	✓ SRRM requires participation of the Product Owner to ensure their goals as agreed in the SPM were met
	X SRRM provides an opportunity for QA to participate in and contribute to the SRRM, but QA may limit their contribution	No QA present in the team	✓ SRRM provides an opportunity for QA to participate in and contribute to the SRRM
	✓ SRRM requires team members to reflect on completed sprint	✓ SRRM requires team members to reflect on completed sprint	✓ SRRM requires team members to reflect on completed sprint
	X SRRM expects team members to identify areas for improvement in future sprints, which rarely occurs	✓ SRRM expects team members to identify areas for improvement in future sprints	X SRRM expects team members to identify areas for improvement in future sprints, which does not always happen

Characteristic of Clan Control: Members exhibit a strong sense of identity with the clan			
Factor	Case C1	Case C2	Case C3
Team members work and cooperate as a team	No customer present in the team	✓ SRRM expects the Product Owner to collaborate with the team to review the software delivered and provide feedback	✓ SRRM expects the Product Owner to collaborate with the team to review the software delivered and provide feedback
	✓ SRRM expects the team members to collaborate and identify areas for improvement which rarely occurs	✓ SRRM expects the team members to collaborate and identify areas for improvement	✓ SRRM expects the team members to collaborate and identify areas for improvement
Team interests emphasised over self-interests	Not evident	Not evident	Not evident

Table 5.3.1-1 SRRM: Members exhibit a strong sense of identity with the clan

5.3.2 Members exhibit a strong sense of commitment to the clan

In section 2.5.4 (p.34), team commitment was described as teams (clans) developing goals and values that members may accept, members exerting considerable effort on behalf of the team (clan), exhibiting a desire to remain a member of the team (clan), identifying with the team, and demonstrating loyalty to the team. These are the five factors that are examined in this section.

Team develop goals and values that members accept

There was evidence to suggest that the SRRM contributed to the development of commitment to the team (Table 5.3.2-1). As discussed in section 5.3.1 (p.210) while the teams had similar goals and values for the meeting the meeting was not always effective nor did it always result in actionable outcomes (Obs C1, C3).

Members exert considerable effort on behalf of the team

There was no evidence in any of the three cases to suggest that the SRRM illustrated a willingness of team members to exert considerable effort on behalf of the team. This may be partially due to the nature of the meeting, which is period of review and reflection for the team after the work of the sprint has been completed. It may also be

due to the lack of value placed in the meeting and the lack of resulting actions from the meeting, particularly in C1 and C3.

Characteristic of Clan Control: Members exhibit a strong sense of commitment to the clan			
Factor	Case C1	Case C2	Case C3
Team develop goals and values that members accept	X SRRM requires participation of the customer to ensure their goals were met as agreed in the SPM, but the customer does not participate	✓ SRRM requires participation of the customer to ensure their goals were met as agreed in the SPM	✓ SRRM requires participation of the customer to ensure their goals were met as agreed in the SPM
	✓ SRRM requires team members to reflect on completed sprint	✓ SRRM requires team members to reflect on completed sprint	✓ SRRM requires team members to reflect on completed sprint
	X SRRM expects team members to identify areas for improvement in future sprints, which rarely occurs	✓ SRRM expects team members to identify areas for improvement in future sprints	X SRRM expects team members to identify areas for improvement in future sprints, which does not always happen
Members exert considerable effort on behalf of the team (clan)	Not evident	Not evident	Not evident
Exhibit a desire to remain a member of the team (clan)	Not evident	Not evident	Not evident
Members identify with the team	See Table 5.3.1-1, p.209		
Members are loyal to the team	See Table 5.3.3-1, p.221		

Table 5.3.2-1 SRRM: Members exhibit a strong sense of commitment to the clan

Members exhibit a desire to remain a member of the team

Similar to the SPM there was no evidence in any of the three cases to suggest that the SRRM had a specific influence over individuals and their desire to remain part of the team. As discussed in section 5.1.2 (p.150) team members indicated that they “*like the way this team work*” (Developer 5, C3), they “*have a very good way as team*” (Scrum Master, C2) since the introduction of their selected agile methodology. Individuals had no desire to work on a team that “*don’t follow agile*” (QA, C3) with a “*preference for agile to the traditional approach*” (Developer 3, C1), but this was due to the agile methodology as a whole rather than specific agile practices.

Members identify with the team

The SRRM contributed to the development of a sense of identification with the team, which was discussed in detail in section 5.3.1 on p.210.

Members are loyal to the team

The SRRM also contributed to the development of a sense of loyalty to the team, which are discussed in greater detail in section 5.3.3 on p.218.

5.3.3 Members develop a strong sense of loyalty to the clan

In section 2.5.4 (p.35) loyalty to an organisation was defined as “*a feeling of affection for and attachment to an organisation*” (Buchanan, 1974), which can also be applied in the context of a team or a group where an individual feels an affection for or an attachment to a team or a group. Individuals who are loyal to a clan or a team develop a feeling of affection for and attachment to the team, identify with the team, develop a sense of belonging and solidarity with the team, team interests are emphasised over self-interests, and develop a strong team spirit. These are the five factors that are examined in this section.

Members feel an emotional attachment to the team/ a sense of belonging to the team

The development of loyalty as a result of the SRRM was limited (Table 5.3.3-1). In C2 and C3 the meeting helped somewhat to develop a sense of attachment to the team and a sense of belonging to the team as it contributed to communication, collaboration and interaction amongst team members in (Obs C3, Reported C2). Similar to the SPM in section 5.1.3 (p.156), the regular interaction and communication at the SRRM helped team members to feel a sense of belonging or attachment to the team members, although this was not as prevalent in C1 as interaction and communication was limited at the SRRM in this team (Obs C1).

In C1 discussion and interaction was limited where each team member, including distributed team members, spoke in turn and only reflected briefly on the previous sprint (Obs C1). It was difficult in C1 to see how this practice could contribute to a sense of belonging or sense of attachment to the team. This was corroborated by the

Project Manager who was of the view that “*I think a lot of the conversation that goes on at the retrospective is stuff that we all kind of know and sometimes I think people are saying stuff just because they have to say something*” (Project Manager, C1). In C3 animated discussions were held between team members in C3 where views and opinions were communicated, listened to, discussed with actions identified (Obs C3).

In C2 the SRRM was used as an opportunity to voice opinions and share views “*you hear what everyone thinks*” (Developer 7, C2). For example, where a developer indicated in the retrospective that they were feeling under pressure “*there’s too much load on me*” (Developer 5, C2) this was handled by the team by sharing out the work in future sprints. Another developer viewed the retrospective as “*good because then you get to say the things that you might have collected during the sprint, but hadn’t an opportunity to air.*” (Developer 1, C2). This developer also felt that where issues identified were resolved this helped to develop loyalty in the team as the feeling was that if they were not resolved “*it just would be oh you can say anything and I don’t care because I won’t do anything*” (Developer 1, C2), which could discourage loyalty to the team. The Scrum Master also welcomed feedback from the team in the SRRM, which the team provided on occasion. Consequently, “*I changed myself a bit and we improved as a team after that*” (Scrum Master, C2). The SRRM was also a forum where differences of opinion were resolved with the team “*not letting it go without bringing it up on the retrospective...and the team try to get a solution to move forward*” (Product Owner, C2). This team appeared to use the SRRM to its full potential in contrast to C1 and C3.

In C1 the SRRM was used to highlight the lack of participation of the customer to the Project Manager “*it’s usually the time where you highlight this type of stuff to your manager*” (Developer 2, C1). There was no evidence that the customer, who was distributed, belonged to team as the customer rarely participated regularly in any of the three practices “*often it’s the case that the off-site customer is just not cooperating... you set a meeting with him and he could just not turn up*” (Developer 2, C1). Instead, it was reported that if customer missed their project deadline “*any of that blame they could delegate to us [the team] they will gladly do*” (Developer 5, C1). While this caused conflict between the team and the customer it also resulted in team members feeling a greater sense of attachment within the team “*so amongst the*

team we are a team" (Developer 5, C1). No similar conflict between the customer and the team was reported in either C2 or C3, which was probably due to the fact that the Product Owner was co-located and regularly communicated and interacted with their team in the SRRM. In C3 in particular, the Product Owner and Management participated in the software demonstration part of the retrospective and provided feedback, both positive and negative, to the team (Obs, C3).

In C1 and C3 the SRRM also facilitated the regular participation of QA in the meeting, which was important from a QA perspective in building relationships with the remainder of the team and developing a sense of belonging to the team. For example, QA faced "*stumbling blocks with our automation and the developers stepped in and helped us with the automation... they have established best practices for our automation... and we [QA] try and encourage individuals to adhere to those [in the SRRM]*" (QA, C3). While the QA resources in C1 did not participate in the interviews they did participate in the SRRM's observed and were "*considered part of the team*" (Business Analyst, C1). Like the SPM, the SRRM helped another distributed team member in C1 to "*feel part of the team*" (Technical Architect, C1), which he believed was partially contributed to by the "*continuous communication between the team*" (Technical Architect, C1) at the SRRM. This was only evident in C1 as this was the only team studied that had distributed team members (Table 4.6.4-1, p.115).

Members identify with the team

The SRRM contributed to the development of a sense of identification with the team, which was discussed in section 5.3.1 on p.210 .

Team interests are emphasised over self-interests

There was no evidence in any of the three cases that the SRRM contributed to team members emphasising the team's interests over their own interests. See section 5.3.1, p.210 for earlier discussion on this point.

Characteristic of Clan Control: Members develop a strong sense of loyalty to the clan			
Factor	Case C1	Case C2	Case C3
Sense of attachment to the team	✓ SRRM requires regular communication and interaction amongst team members	✓ SRRM requires regular communication and interaction amongst team members	✓ SRRM requires regular communication and interaction amongst team members
	No customer present in the team	✓ SRRM facilitates regular participation of the Product Owner	✓ SRRM facilitates regular participation of the Product Owner
	✓ SRRM facilitates regular participation of QA in the team	QA not present in the team	✓ SRRM facilitates regular participation of QA in the team
	✓ SRRM facilitates regular participation of distributed team members	No distributed team members present in the team	No distributed team members present in the team
Sense of belonging and solidarity with the team	✓ SRRM requires regular communication and interaction amongst team members	✓ SRRM requires regular communication and interaction amongst team members	✓ SRRM requires regular communication and interaction amongst team members
	No customer present in the team	✓ SRRM facilitates regular participation of the Product Owner	✓ SRRM facilitates regular participation of the Product Owner
	✓ SRRM facilitates regular participation of QA in the team	QA not present in the team	✓ SRRM facilitates regular participation of QA in the team
	✓ SRRM facilitates regular participation of distributed team members	No distributed team members present in the team	No distributed team members present in the team
Sense of identity with the team	See Table 5.3.1-1, p.209		
Team interests are emphasised over self-interests	Not evident	Not evident	Not evident
Strong team spirit	No customer present in the team	✓ SRRM requires the team to demonstrate the completed software to the customer	✓ SRRM requires the team to demonstrate the completed software to the customer
	✓ SRRM requires team members to reflect on the completed sprint	✓ SRRM requires team members to reflect on the completed sprint	✓ SRRM requires team members to reflect on the completed sprint
	X SRRM expects team members to identify areas for improvement in future sprints, which rarely occurs	✓ SRRM expects team members to identify areas for improvement in future sprints	X SRRM expects team members to identify areas for improvement in future sprints, which does not always happen

Table 5.3.3-1 SRRM: Members develop a strong sense of loyalty to the clan

Members develop a strong team spirit

As discussed in section 5.1.3 (p.156) a strong team spirit existed in C2 and C3 with mixed views of the strength of the team spirit present in C1. The SRRM may have

contributed to team spirit in each case due to the regular involvement of team members in the demonstration of the completed software to the customer (C2, C3), the opportunity to reflect on work completed in the sprint (C1, C2, C3) and the opportunity to identify areas for improvement in future sprints (C1, C2, C3). As the SRRM was not used to its full potential in C1 this may be a contributing factor to the mixed feelings on team spirit. As areas for improvement are rarely identified in C1, no action was taken to make improvements or changes for the benefit of the team.

5.3.4 Members develop common interests and goals which evolve as tasks progress

In section 2.5.4 (p.36) detail was provided on the development of common interests and goals in a clan. Where clan control exists goals are typically unknown at the outset. Instead, clans develop common interests and goals, which evolve as tasks progress and as members become socialised into the clan and become familiar with the norms and values of the clan. In addition, where clan control exists in ISD teams they have autonomy over their goals. These two factors are examined in this section.

Common interests and goals develop, which evolve as tasks progress

There was no evidence in C1 that the SRRM was used to develop common interests and goals for the team. This may be because the SRRM was not considered to “*add a whole lot of value...as a lot of the conversation that goes on at the retrospective is stuff that we all kind of know and sometimes I think people are saying stuff just because they want to*” (Project Manager, C1) or “*people tend to say a few things alright but it's a bit like 'here we go again'*” (Developer 1, C1). Reflection on the sprint was conducted by team members, but it was brief and there was little identification of areas for future improvement. However, there were plans in place to “*change our retrospectives*” (Project Manager, C1) in attempt to add more value to them.

In C2 the SRRM had value and there was a common goal and interest in the team to reflect and “*analyse how we worked, sometimes we tend to forget what we have learned, sometimes we have remembered it but I think it does help us to focus on this we have seen before, if this is a pattern, we will not go there again.*” (Developer 1, C2). In this instance the goal or interest of the team was to avoid repeating the same

mistakes. This was also evident in C3 where the team “*discussed what went well and what went not so well, see what we can improve and come up with new ideas*” (Scrum Master, C3). However, another team member felt that the SRRM “*could be really useful....we have discussions with each other which are very interesting, but then there are no actions behind them*” (Developer 1, C3). A second developer felt that “*if it is just us, it tends to be of value, people will talk and stuff...there have been a few times where slightly higher up individuals [management] have come and then everybody else clams up and doesn't say anything*” (Developer 2, C3). Two SRRM's were observed in C3 (Table 4.7-1) and discussions were observed between team members with some actions identified; for example, an action was identified “*to get 80% of the coverage in unit test of the code*” (Developer 3, C3).

C2 also had a common goal and interest on how to address a difficulty with completing long-term or complex tasks within a sprint, which they have “*talked about on the retrospective and I think I'm not the only one who feels that way and the problem is how to manage it of course*” (Developer 4, C2). The SRRM was used by the team to try and identify a solution to the problem, although it was acknowledged that it was not an urgent goal as “*right now there are no larger design works*” (Developer 4, C2). This was not experienced in C1 or C3 as there was no evidence of difficulty with incorporating long-term tasks into a sprint.

Characteristic of Clan Control: Members develop common interests and goals which evolve as tasks progress			
Factor	Case C1	Case C2	Case C3
Common interests and goals develop, which evolve as tasks progress	✓ SRRM requires team members to reflect on completed sprint	✓ SRRM requires team members to reflect on completed sprint	✓ SRRM requires team members to reflect on completed sprint
	X SRRM expects team members to identify areas for improvement in future sprints, which rarely occurs	✓ SRRM expects team members to identify areas for improvement in future sprints	X SRRM expects team members to identify areas for improvement in future sprints, which does not always happen
Teams have autonomy over goals and how to achieve them	Not evident	✓ SRRM provides team members with autonomy to action items for improvement	✓ SRRM provides team members with autonomy to action items for improvement

Table 5.3.4-1 SRRM: Members develop common interests and goals which evolve as tasks progress

Team members have autonomy over goals and how to achieve them

As stated earlier one of the aims of this meeting was to identify areas for improvement in future sprints with teams having autonomy over how to achieve these goals and could decide when and how to action them, which became goals for the team. This occurred in C2 and C3, whereas none were identified in C1.

5.3.5 Rituals and ceremonies identify and reinforce acceptable behaviours

Section 2.5.4 (p.38) described rituals and ceremonies as regular occurrences that “*reinforce the implicit, but well understood codes of correct conduct*” (Macintosh, p138). They are a source of information, communicate values and beliefs to members, require the recurring collective participation of members, require members to behave in a manner that is consistent with agreed-upon behaviours, and can be in the form of rules and regulations that are considered acceptable by the clan. This section discusses the ritual and ceremony that takes place during the SRRM and also how the SRRM identifies and reinforces acceptable behaviours.

Rituals and ceremonies

In ISD projects, rituals and ceremonies occur in many ways such as meetings. In ASD the SRRM is a third such meeting. This meeting took place on a regular basis at the end of each sprint. It required team members to comply with certain rules and regulations, it reinforced certain behaviours to which team members agreed in order to be part of the ASD team, and it was a source of information for team members where the values and beliefs of the team could be communicated (Table 5.3.5-1). This section first describes the how each of the three teams conducted the ritual and ceremony of the SRRM. While there were some similarities across the three cases, each had their own nuances, which were a consequence of a number of factors, such as the environment they worked in and the distribution of the team. This is then followed by detail on the particular behaviours that were considered acceptable for each team as a result of using the SRRM. It is not intended to be an exhaustive list of behaviours, but to identify specific behaviours that have resulted in the three teams studied as a consequence of using the SRRM. These behaviours were identified through observations and responses from interviewees. There were variations across each of the three teams as to what was considered acceptable behaviour. It was

possible that the SRRM encouraged other acceptable behaviours, which were not identified because interviewees did not consider them important, or where something occurred so regularly that they were not consciously aware of their behaviours or any subtleties of interactions between themselves and other team members. Additional observations of the teams over a sustained period of time could identify additional behaviours. This section concludes with an examination of how the SRRM reinforced those behaviours.

The SRRM was **conducted on a regular day, at the same time in the same location**. These days and times varied across the three cases to suit the requirements of the team and the environment within which the team operated. In C1 the SPM was conducted at 2pm on a Friday afternoon (Obs C1) in a conference room, which allowed distributed team members to participate via a conference call. All team members were expected to participate in the meeting but as reported in section 5.3.1 (p.210) members did not always participate. Due to the time selected this required team members in India to remain at work for several hours in order to attend the meeting, which was necessary once every three weeks. In C2 the meeting was not observed, but it was reported by the Scrum Master that this meeting was held on a Friday afternoon at the end of each sprint. In C3 the SRRM was conducted at 3pm on the Friday afternoon at the end of each sprint and was held in the only large meeting room in the organisation

According to the methodology the duration of the meeting should be time-boxed to a number of hours, although this can vary (section 3.3.4, p.68). In all three cases the **duration of the meeting varied from one sprint to the next**, but it always concluded within a single afternoon. The duration of the meeting depended on the length of the software demonstration, where conducted and the level of discussion between team members. In C1 the meeting observed had a duration of approximately 15 minutes with limited discussion (Obs C1). No demonstration of the software was conducted in C1 as the customer was not present. Several reasons for this were detailed earlier in section 5.1.1. In C3 the SRRM lasted approximately one hour, which consisted of a demonstration of the software to management and the Product Owner for approximately 30 minutes (Obs C3). This was followed by the

retrospective part of the meeting, which also had a duration of approximately 30 minutes (Obs C3). The meeting was not observed in C2.

Based on observations **each meeting followed its own individual process** in the two teams observed. In C1 the meeting was very short in duration (15 minutes), which was facilitated by the Project Manager. All team members participated including distributed team members. The Project Manager asked each team member in turn to comment on their work in the previous sprint to which each team member complied (Obs C1). Feedback in the two SRRM's observed was predominantly positive with no recommendations for improvement in future sprints. However, the Project Manager did provide some specific feedback for improvement to the team in the second retrospective meeting observed (Obs, C1). Following the completion of the meetings a separate informal discussion was held with the Project Manager in relation to the SRRM. The Project Manager detailed that these meetings are the one opportunity for all team members to participate in a team meeting and air any issues that they may have, or to identify any possible delays/impacts that one task might have on another. He reported that some SRRM's could get quite heated depending on which stage of the project the meeting is held. However, recent meetings were non-eventful, with almost all issues known in advance of the meeting, which may be related to the nature of the current workload, which was predominantly maintenance fixes, rather than core software development.

In C3 the SRRM was divided into two parts, the first part consisted of a demonstration of the software to the Product Owner and to senior management, which was led by one of the developers (Obs, C3). This person rotated in each meeting as "*I try to ask everybody to take part in the demo*" (Scrum Master, C3), which was intentional as the Scrum Master felt that it improved the communication skills of team members. This was corroborated by QA "*we allow people to develop personally by allowing different people to do the demonstrations*" (QA, C3). After the demonstration completed various questions were asked by the senior management team and the Product Owner to which the team responded. The Product Owner took notes on a number of items that he needed to remember or clarify later (Obs, C3). The retrospective was held directly after the demonstration in the same meeting room and was conducted by the team and did not include the Product Owner

or senior management. (Obs, C3). Team members discussed any difficulties that they had in the sprint completed and how they may be addressed going forward. They also discussed any potential issues that may arise in the next or future sprints (Obs, C3).

Rituals and ceremonies identify acceptable behaviours

In each case team members had autonomy to develop their rules and regulations for the SRRM and they also identified acceptable behaviours for the team. Few behaviours were identified in the SRRM, partially due to the short duration of one of the meetings (C1), and partially due to the lack of observation of another (C2). Few rules, regulations and behaviours were determined in C1, which was due to the short duration of the meeting and the lack of engagement at the meeting. For example, all team members were expected to attend the SRRM, but in C1 it was an acceptable behaviour not to attend the SRRM, whereas in C2 and C3 it was unacceptable to be absent from the meeting. Similarly, in C1 the meeting was chaired by the Project Manager, team members were expected to be punctual for the meeting, and they were expected to reflect individually on the work they had completed in the sprint (Obs C1). Team members rarely identified actions for improvement, which was an acceptable behaviour in the team and was not addressed by team members at the SRRM.

In C2 and C3 team members were expected to be punctual for the SRRM and to contribute to the SRRM in turn, to which they complied (Reported C2, Obs C3). But, additional behaviours were identified in C2 and C3. In both C2 and C3 the Product Owner “*attends the demo part of the meeting*” (Product Owner, C3) to review the software delivered, provide feedback, and identify any “*observations*” (Product Owner, C2). In C3 management also regularly attended the demonstration to monitor the progress of the team and provide feedback (Obs C3). It also helped “*to clearly set their expectations about when the next release will be ready*” (Product Owner, C3). In C3 different developers “*do the [software] demonstrations... and the task is rotated*” (QA, C3). In both C2 and C3 “*every single person in the team can give their opinion about how they thought things had gone*” (QA, C3), individuals can “*talk*

about how we work together... and can give feedback" (Scrum Master, C2) to help identify areas for improvement.

Rituals and ceremonies reinforce acceptable behaviours

In C1 the SRRM did not always reinforce acceptable behaviours as team members did not attend all the time, the customer did not participate and the team did not demonstrate the completed software. Little activity occurred at this meeting in C1. In C2 and C3 the SRRM required regular attendance at and participation at the meeting, which helped to reinforce the rules, regulations to team members. As the meeting was not observed in C2 it was difficult to determine if the SRRM reinforced any behaviour in this team. In C3 any developer could demonstrate the software to the Product Owner, but this was reinforced by the Scrum Master who requested that the person "*who shows something in the demo [changes] from sprint to sprint*" (Scrum Master, C3). Also, the Product Owner and anyone from the management team could attend the SRRM in C3, which they did frequently. They used this forum to determine the progress of the team, provide feedback, and make suggestions for future releases of the product.

Characteristic of Clan Control: Rituals and Ceremonies identify and reinforce acceptable behaviour			
Factor	Case C1	Case C2	Case C3
Rituals and Ceremonies	✓ SRRM takes place at the same time in the same location at the start of each sprint	✓ SRRM takes place at the same time in the same location at the start of each sprint	✓ SRRM takes place at the same time in the same location at the start of each sprint
	✓ SRRM concludes within a specified timeframe	✓ SRRM concludes within a specified timeframe	✓ SRRM concludes within a specified timeframe
	✓ Each SRRM meeting follows a similar process	✓ Each SRRM meeting follows a similar process	✓ Each SRRM meeting follows a similar process
Rituals and ceremonies identify acceptable behaviour	✓ SRRM provides the team with autonomy to develop rules and regulations for the meeting e.g. rarely identified actionable items	✓ SRRM provides the team with autonomy to develop rules and regulations for the meeting e.g. Product Owner attends the meeting	✓ SRRM provides the team with autonomy to develop rules and regulations for the meeting e.g. meeting is always chaired by the same individual
	✓ SRRM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance	✓ SRRM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance	✓ SRRM provides the team with autonomy to identify acceptable behaviours for the meeting e.g. punctual attendance

Characteristic of Clan Control: Rituals and Ceremonies identify and reinforce acceptable behaviour			
Factor	Case C1	Case C2	Case C3
Rituals and ceremonies reinforce acceptable behaviour	No customer present in the team	✓ Product Owner attends the SRRM to provide review and feedback	✓ Product Owner and management attend the SRRM to provide review and feedback
	Completed software is not demonstrated to the customer	This meeting was not observed	✓ Any developer can demonstrate the completed software at the SRRM
	✓ Team members can identify areas in which the team can improve at the SRRM, which does not occur	✓ Team members can identify areas in which the team can improve at the SRRM	✓ Team members can identify areas in which the team can improve at the SRRM
Rituals and ceremonies reinforce acceptable behaviour	X SRRM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour, but team members do not always attend	✓ SRRM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour	✓ SRRM requires regular attendance at and participation in the meeting to reinforce acceptable behaviour
	No customer present in the team	This meeting was not observed.	✓ Product Owner and management provide feedback to the team at the SRRM
	No customer present in the team	This meeting was not observed.	✓ Developers rotate the task of demonstrating the completed software to the Product Owner from one SRRM to the next

Table 5.3.5-1 SRRM: Rituals and ceremonies identify and reinforce acceptable behaviours

5.3.6 Members are influenced by shared norms, values and beliefs

As detailed in section 2.5.4 (p.39) members of a clan are influenced by shared norms, values and beliefs that exist within the clan with the clan correcting any deviant behaviour. Clan control is in operation when behaviour is influenced by shared norms, values and beliefs.

In all three cases the SRRM resulted in the development of certain norms, values, and beliefs due to its ritual described in section 5.2.5 (p.194), which have somewhat influenced the behaviours of team members (Table 5.3.6-1). This section first describes the how shared norms of the team influenced behaviour, which is followed by how shared values and beliefs influenced the behaviour of team members in each

of the three cases. This was not intended to be an exhaustive list of shared norms, values and beliefs, but to identify specific examples that influenced behaviour in the three teams studied as a consequence of using the SRRM. These behaviours were identified through observations and responses from interviewees. It was possible that other norms, values, and beliefs were present, which were not identified for the same reasons discussed in section 5.3.5 (on p.224).

Members are influenced by shared norms

The shared norms of the SRRM influenced the behaviour of team members in different ways. In C1 the norm was to briefly reflect on work completed for the sprint and not to identify areas of improvement for the team for future sprints (Obs C1). As team members did not deviate from this norm this resulted in an ineffective SRRM and it was not surprising that team members felt like they "*don't get the benefit out of that [retrospective]*" (Developer 3, C1), which also explained the lack of engagement observed at the meeting (Obs C1). In C3 there was plenty of discussion at the SRRM and some actions were identified (Obs, C3), although the Scrum Master felt that many times "*there is too much to talk or there are no very specific actions*" (Scrum Master, C3) and "*it is not as useful as it should be*" (Developer 1, C3). It was also acceptable at this meeting for the Product Owner and management to attend the retrospective part of SRRM where team members discussed what went well and what was done poorly, but this influenced the behaviour of the team as "*nobody wanted to talk then*" (Developer 2, C3). However, management "*are lately no longer attending [the retrospective part]..and it is much more beneficial*" (Developer 2, C3).

There was an expectation in each of the three teams that everybody would "*attend*" (Developer 3, C3) and "*contribute to*" (Developer 2, C2) the SRRM to which team members in C2 and C3 conformed. As discussed in section 5.3.1 (p.210) team members in C1 did not always conform to this norm. The lack of contribution in C1 also resulted in team members not taking the opportunity to question when the team did not perform as expected. For example, where a task was not completed and it was "*rolled into the next sprint*" (Developer 3, C1), the team rarely "*questioned why it wasn't completed*" (Developer 3, C1). In C2 it was the norm was to contribute and

communicate difficulties team members were comfortable stating if “*they had a very high workload*” (Developer 5, C2) and others willingly agreed to help and take “*the load off your shoulders... by taking that into account in the following sprint planning meeting*” (Developer 5, C2). Similar evidence was not found in C1 or C3.

In C1 and C3 it was the norm for QA to attend the SRRM, but in C3 QA used the SRRM to highlight to the team how to improve from a quality perspective “*we give hints on how to improve things, which we would reinforce in a retrospective to try and encourage individuals to adhere to those.*” (QA, C3). This was not observed or reported in C1 even though QA participated in the SRRM. This may be due to the cultural differences between the QA team, who were based in India and the rest of the team, who were predominantly located in Ireland. It was reported that the QA team were likely “*to give a more positive picture*” (Project Manager, C1) than what was happening in reality as they tended to be “*submissive*” (Developer 2, C1) and they did not really question other team members. C2 did not consist of any QA team members, so this was not observed or reported in this team.

Members are influenced by shared values and beliefs

The SRRM had a limited contribution to the development of shared values and beliefs amongst team members. The shared values and beliefs of learning, honesty, cooperativeness, communication, and accountability as detailed in section 5.1.6 (p.169) were only partially reinforced by the SRRM as it was not used to its full potential in at least one of the cases (C1). In C1 team members communicated with each other, but learning was limited due to the poor implementation of the practice, whereas learning and communication was promoted in C2 and C3 where the SRRM was conducted more effectively and team members cooperated with each other to identify solutions to difficulties. The Product Owner felt that it is not possible to hide “*from the willingness to learn as everyone is exposed [in what they do]*” (Product Owner, C3). Teams in C2 and C3 demonstrated some level of accountability to the team by identifying difficulties or lessons learned and developing actions to address these difficulties, although in C3 these were not always achieved, which was not done in C1. The SRRM also encouraged team members to be honest in their reflection on

the work completed in the previous sprint, but as little was reported (Obs C1, C3) it is unclear if the SRRM contributed to this value.

Characteristic of Clan Control: Members are influenced by shared norms, values and beliefs			
Factor	Case C1	Case C2	Case C3
Shared norms influenced behaviour	X Used by the team to reflect on the sprint with few actionable items identified, resulting in lack of engagement at the SRRM	✓ Used by the team to reflect on the sprint with areas for improvement identified	✓ Used by the team to reflect on the sprint with some actionable items identified
	No management present at the SRRM	No management present at the SRRM	X Attendance of management at the SRRM discouraged team members from providing feedback
	X Team members are expected to regularly attend the SRRM, but they may vary their levels of attendance	✓ Team members are expected to regularly attend the SRRM and they always attend	✓ Team members are expected to regularly attend the SRRM and they always attend
	X Team members are expected to contribute to the SRRM in addition to their expected contribution, but levels of contribution can vary	✓ Team members are expected to contribute to the SRRM in addition to their expected contribution, but levels of contribution can vary	✓ Team members are expected to contribute to the SRRM in addition to their expected contribution, but levels of contribution can vary
	Not evident	✓ Team members openly ask for assistance and a reduced workload in the next sprint	Not evident
	X QA rarely question team members or report difficulties	No QA present in the team	✓ QA highlight areas for improvement, which the team take on board in future sprints
Shared values and beliefs influenced behaviour	X SRRM encourages learning, which influences behaviour e.g. team members are poor at identifying areas for improvement, resulting in no learning	✓ SRRM encourages learning, which influences behaviour e.g. team members identify areas for improvement, resulting in increased learning	✓ SRRM encourages learning, which influences behaviour e.g. team members identify areas for improvement, resulting in increased learning
	X SRRM encourages communication, which influences behaviour, but communication and interaction was limited	✓ SRRM encourages communication, which influences behaviour, resulting in discussions on how the team could improve	✓ SRRM encourages communication, which influences behaviour, resulting in discussions on how the team could improve
	✓ SRRM encourages team members to be honest in their reflection on the work completed	✓ SRRM encourages team members to be honest in their reflection on the work completed	✓ SRRM encourages team members to be honest in their reflection on the work completed

Characteristic of Clan Control: Members are influenced by shared norms, values and beliefs			
Factor	Case C1	Case C2	Case C3
	X SRRM encourages cooperation amongst team members to identify solutions to difficulties identified, but no difficulties are identified	✓ SRRM encourages cooperation amongst team members to identify solutions to difficulties identified	✓ SRRM encourages cooperation amongst team members to identify solutions to difficulties identified
	X SRRM encourages accountability amongst team members to report on difficulties encountered, but none were reported	✓ SRRM encourages accountability amongst team members to report on difficulties encountered	✓ SRRM encourages accountability amongst team members to report on difficulties encountered

Table 5.3.6-1 SRRM: Members are influenced by shared norms, values and beliefs

5.3.7 Members evaluate and reward or sanction each other

In section 2.5.4 (p.40) it was identified that clans evaluate, reward and sanction each other on their conformance to the agreed norms, values and goals of the clan.

The SRRM provided team members with ample opportunity to evaluate and reward or sanction each other, although the evidence obtained related mostly to conformance to goals of the team (Table 5.3.7-1). It also provided an opportunity for management or the customer to demonstrate their appreciation to the team where agreed goals were achieved. While formal rewards were evident across all three cases, only two of the three cases (C2 and C3) used the SRRM as a forum to reward the team.

Members evaluate each other on their conformance with the team's norms, goals, and values

The SRRM expected teams to evaluate whether they fulfilled the goal as defined in the SPM. This was conducted briefly in C1 (Obs C1), but if all tasks were not achieved they were subsequently “rolled into the next sprint” (Developer 3, C1) with few questions asked as to “why it wasn’t completed” (Developer 3, C1). This meeting was not observed in C2, but team members did evaluate their conformance to the common goal as they have seen sprints “fail a lot the last five or six sprints” (Developer 6, C2), yet there were no repercussions for the team. In C3 the goal for

the sprint was reiterated at the start of the SRRM and team members reported individually whether they had completed their tasks as expected (Obs C3), although evidence was also obtained from the web-based Scrum Board. Similar to C1 where tasks were incomplete they were moved to subsequent sprints.

Members reward each other for conformance to the team's norms, goals, and values

There was no evidence in C1 that members rewarded each other for conformance to the norms, values, and beliefs of the team. In this team rewards were formal and related to money “*it’s monetary really....pretty much yeah people go for their bonus at the end of the year*” (Developer 1, C1). Other rewards were driven mainly by management and were not part of the SRRM; for example, “*a pat on the back when you hit the deadline*” (Developer 1, C1), recognition internally within the organisation “*we got a lot of kudos last yearwe were getting plaudits for the work we delivered*” (Developer 2, C1), an email from management to senior management congratulating the team on a successful release “*he will copy in his boss and his bosses boss and a few levels above that, so a lot of people will know*” (Business Analyst, C1). This may be attributable to the hierarchical structure of this organisation and a desire for the organisation in Ireland to demonstrate their value to head office in the United States as software development is off-shored to the Irish office. The lack of rewards or sanctions in the SRRM may also be caused by the lack of value team members placed in meeting itself as discussed in section 0, p.229.

This contrasted with C2 and C3 where the SRRM was regularly used to recognise the efforts of the team and verbally reward and praise the team; for example when goals were achieved, “*the retrospective provides you with feedback or praise that you have done a good job*” (Developer 7, C2), “*we get to thank people... and that comes from the Scrum Master*” (Developer 2, C2), or “*we praise each other sometimes during the demos*” (Scrum Master, C3). As both of these teams conducted a demonstration of the completed software to the Product Owner, and management at the start of the SRRM this provided an opportunity for these groups to provide feedback to the team on the delivery of agreed goals (Obs C3). In C3 the Product Owner and management regularly praised the team in these meetings “*we receive*

verbal praise from management “well done team”, “Good job team” in the retrospective” (Developer 1, C3). A congratulatory email may also be communicated to the whole company “*especially when projects are completed successfully*” (Developer 4, C3). This was not something that was ever done in C2 “*I have some trouble imagining what kind of accomplishment that would be to trigger such a reaction [the sending of an email]*” (Developer 1, C2). However, if management were not happy with progress this would also be indicated to the team. For example, “*if the director of product development happens to be in a demo and he is not happy he will say it.*” (QA, C3). Positive feedback was provided by the customer occasionally in C2 “*if during the demo you get some positive comments that's' a good grade.*” (Developer 1, C2). In C2 the Product Owner also used this meeting to demonstrate the customer’s appreciation to the team “*the only thing I can do as a reward is more like tell the team how outside stakeholders appreciate what the team do....It happens when we know that they are achieved very good...I would say once per three sprints*” (Product Owner, C2)

There was some degree of evidence in C2 and C3 that recognition was not always bestowed on team members where a significant personal contribution was made to a project “*If one person makes a large contribution to the success of a project, they are not recognised in any way*” (Developer 1, C2). A developer in C3 thought there may be some personal rewards for making an outstanding contribution to the team, but was unsure “*If there were personal rewards, they are private*” (Developer 1, C3). Another team member felt that the team “*don't really self-reward.*” (QA, C3). There was no evidence in C1 that substantial individual contributions to the team were recognised. A lack of recognition of the efforts of team members may stem from the environment that team members were traditionally familiar with, where rewards were identified and acknowledged by a manager at an end of year performance review. These formal performance reviews still existed in each of the three cases, although C3 were changing their performance review system where “*certain goals [for each individual] are group goals and that wouldn't have been the case before*” (QA, C3) with up to “*60% or even more of it is based on the team performance and there is an individual part based on your own performance*” (Scrum Master, C3). This new rewards system was coming into place within a few months.

Characteristic of Clan Control: Members evaluate and reward or sanction each other on their conformance to the agreed norms, goals, and values			
Factor	Case C1	Case C2	Case C3
Evaluation of conformance to agreed norms, goals and values	✓ Teams evaluate tasks completed against the goal defined in the SPM	✓ Teams evaluate tasks completed against the goal defined in the SPM	✓ Teams evaluate tasks completed against the goal defined in the SPM
	Not evident	✓ Allows teams to self-reflect and identify areas for team improvement	✓ Allows teams to self-reflect and identify areas for team improvement
Reward for conformance to agreed norms, goals and values	Not evident	✓ Provides rewards for conformance to the agreed norms, values and goals of the clan.	✓ Provides rewards for conformance to the agreed norms, values and goals of the clan.
	Not evident	✓ Teams verbally praise each other for goals achieved	✓ Teams verbally praise each other for goals achieved
Sanction for non-conformance to agreed norms, goals and values	Not evident	✓ Sanctioned for non-conformance to agreed goals	Not evident

Table 5.3.7-1 SRRM: Members evaluate and reward or sanction each other

Members sanction each other for non-conformance to the team's norms, goals, and values

There was limited evidence of team members sanctioning each other in the SRRM in any of the three cases. The only evidence identified was in C3 where teams were sanctioned for non-conformance to the agreed goals of the sprint. Management in C3 verbally expressed dissatisfaction with the team if they were not satisfied with progress. Even though in C2 the team failed to complete several sprints in a row “*we have seen our sprints actually fail a lot the last five or six sprints.*” (Developer 6, C2), which was reported as due to poor estimation “*it started out very well because we did the easy stuff so to speak or every time we think it’s only four hours left but then when digging deep down into a problem you realise that oh it’s maybe eight hours and then the sprint planning was wrong from the beginning*” (Developer 6, C2) there did not appear to be any repercussions from their management. However, this had frustrated the Product Owner and while he did not want productivity to decrease “*maybe we need to go there to take in to underestimate totally how much we can do...so, they feel as a success and they can show off that we met what we have*

expected and get into a good spiral instead of the bad one where always end up taking too much and we will always fail" (Product Owner, C2) he felt that he did not "*have the authority to stand up in the sprint to tell the team how they should work*" (Product Owner, C2) and that instead this should be addressed by the Scrum Master. However, the Product Owner reported that in the SRRM the previous week the team had finally "*started to address this problem to see how we shall handle those situations*" (Product Owner, C2).

5.4 Summary

This chapter presented the results from each of the three teams under each component of the conceptual framework, Figure 3.5-1 (p.77). A summary of the findings is presented in Table 5.4-1, which demonstrates where the agile practices helped to achieve each resultant characteristic of clan control. The table uses a checkmark (✓) to indicate where an agile practice showed evidence of achieving a resultant characteristic of clan control. An 'M' indicated where there was mixed evidence within a team. Where there was no evidence of achieving the resultant characteristic of clan control this is indicated by a '--' in the corresponding column. With the exception of the characteristic '*Members evaluate and reward, or sanction each other*' all other characteristics were observed either fully, or to some extent in each of the cases studied. In the table below the final three characteristics were presented in a number of distinct rows as the label for the characteristic lends itself to separation for clarity.

	Sprint Planning Meeting			Daily Scrum Meeting			Sprint Review and Retrospective Meeting		
Case	C1	C2	C3	C1	C2	C3	C1	C2	C3
Clan Control Characteristic									
Members exhibit a strong sense of identity with the clan	M	✓	✓	M	✓	✓	M	✓	M
Members exhibit a strong sense of commitment to the clan	✓	✓	✓	✓	✓	✓	M	✓	M
Members develop a strong sense of loyalty to the clan	M	✓	✓	✓	✓	✓	M	✓	M
Members develop common interests and goals	✓	✓	✓	✓	✓	✓	M	✓	M
Rituals and ceremonies identify acceptable behaviour	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rituals and ceremonies reinforce acceptable behaviour	M	✓	✓	M	✓	✓	M	✓	✓
Members influenced by shared norms	M	✓	✓	M	✓	✓	M	✓	M
Members influenced by shared values and beliefs	M	✓	✓	✓	✓	✓	M	✓	✓
Members evaluate each other on their conformance to agreed norms, goals, and values	--	✓	--	✓	✓	✓	--	✓	✓
Members reward each other on their conformance to agreed norms, goals, and values	--	--	--	--	✓	-	--	✓	✓
Members sanction each other on their conformance to agreed norms, goals, and values	--	✓	--	--	✓	✓	--	--	--
M = Mixed Evidence -- = Not evident									

Table 5.4-1 Summary of Findings

In the next chapter recommendations are provided on how to improve the agile practices to achieve higher levels of clan control in ASD teams. The subsequent chapter (section 7.2, p.268) examines the findings from this study as summarised in Table 5.4-1 in the context of the literature discussed in Chapter 2 and Chapter 3, and will detail their significance and implications for clan control in ASD teams.

6 Recommendations

6.0 Introduction

The purpose of this chapter is to answer the third research objective. It presents a set of recommendations that detail how agile practices in each of the three cases may be improved to achieve higher levels of clan control. These recommendations are based on the findings from this study. As each ASD team tailored the agile practices to suit their specific needs the recommendations illustrate where each of the teams exhibited good practices (those that had a positive impact on the team) and where poor practices were implemented (those that had a negative impact on the team) as reported by the participants of the study ~~and did not conform to the prescribed methodology~~. These recommendations are not intended to be an extensive list as they are based solely on the findings from three ASD teams and how these teams have implemented the agile practices in their particular setting. However, they are a list on which future research can build to develop a more comprehensive set of recommendations. The first section (6.1) presents the recommendations for the sprint planning meeting. This is followed by recommendations for the daily scrum meeting (section 6.2) and the sprint review and retrospective meeting (section 6.3).

6.1 Sprint Planning Meeting Recommendations

In Table 5.4-1 on p.238, the SPM helped to achieve six of the seven resultant characteristics in each of the three cases either wholly or partially, which suggested that the SPM can contribute to the development of clan control. Each of the cases studied had their own unique setting and implemented the SPM in different ways. Some practices were implemented well, whereas others were poorly executed by the teams, or poor practices had possibly developed over time. Both good and poor practices are discussed with recommendations made where the practice was either strong or weak in particular teams (Table 6.1-1, p.247). These are indicated in the tables by providing the label for the case(s) after each recommendation. Each recommendation is labelled with the letter 'R' followed by a number in increasing order. Note that the same recommendation may correspond to a number of different resultant characteristics of clan control.

R1: Ensure continued participation and contribution of all team members in the SPM and allow everyone to speak freely

Where team members participated in and contributed to the SPM it helped them to develop their identity within the team, develop a sense of loyalty to the team, and allowed them to be influenced by the shared norms, values and beliefs that existed within the team, resulting in compliance with acceptable behaviours as defined by the team. Teams in C2 and C3 adhered to this good practice and this was evident in both of these teams. As both these teams had implemented Scrum in recent months it was likely that all team members in these teams considered this important in order to become familiar with each other, the practice, and to ensure they could deliver on the agreed goal for the sprint. This was poorly implemented in C1, yet no repercussions were recounted by the team for a lack of participation or contribution. This team were using the practice for two years on the same project and they may have become lenient with each other and their requirement to attend, even though it was reported by team members present that the reason for non-attendance was due to meeting conflicts. The importance of these other meetings was not communicated. Team members may also be influenced by the behaviours of other team members if non-attendance or a lack of contribution had developed as a norm. Their lack of attendance and contribution when present could indicate apathy towards the team, resulting in a lack of loyalty or sense of identification with the team. The Project Manager should re-iterate the importance of the meetings to the team and encourage team members to minimise their absence at the meetings.

R2: Ensure participation of new and distributed team members in the SPM from the outset and allow everyone to speak freely

The introduction of new and distributed team members to the SPM from the outset proved valuable for these individuals. It quickly integrated them into the team where they became familiar with team members, the norms of the team, the behaviours considered acceptable by the team, the nature of the project, the work that was under development, and ensured that they communicated and interacted with the team immediately, even if they had little to say at the start. All of this helped to develop their sense of identity within the team and develop their sense of loyalty to the team. This practice was enforced in both C1 and C2 where new and distributed team

members were introduced to the team. In C2 the team made a concerted effort to integrate a new team member in the SPM by providing an introductory activity (unrelated to the project) for the new team member to complete, so they could quickly participate in the ritual of the SPM and become familiar with the norms of the practice. In C1 some efforts were made to conduct site visits between Ireland and the USA to assist with relationship building between the two teams and the integration of distributed team members into the team. However, no site visits were reported between team members in India and the other two distributed sites. C1 used a conference phone to communicate with distributed team members, which caused difficulties with comprehending conversations due to the poor quality of the phone line and language barriers. Communication and interaction between team members could be improved through the use of technologies such as video-conferencing, which were available for use within the organisation. It may also help to build relationships and encourage team members to improve their engagement during the SPM if they known that distributed team members can observe them.

R3. Facilitate QA participation in the SPM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team

QA were regularly present in the SPM in both C1 and C3, which helped to develop a loyalty between them and the remainder of the team that was not present previously (as reported in C3). It also helped QA to develop their identity within the team, and allowed them to influence the shared norms, values and beliefs that existed within the team by vocalising the needs of QA and ensuring that team members were aware of how to improve their software delivered from a quality perspective. This good practice was more prevalent in C3 than in C1, which was partially due to the distributed nature of the QA team in C1, the frequent change in QA personnel, as well as the cultural differences between the QA team and the development team. The cultural difficulties with QA in C1 could be addressed to some extent by site visits between the two countries. However, this team are resolving the difficulties with QA by re-locating the QA function to Ireland where they will work alongside the core software development team.

R4. Encourage and facilitate customer participation in the SPM

Customer representation was present in C2 and C3 in the form of the Product Owner, who was co-located with the team. Active participation in the SPM by the customer helped to build relationships with the remainder of the team, ensured their goals were aligned with those of the team, that prioritisations of tasks were correct, and that they became familiar with the norms and values of the team. It also helped the customer to develop a sense of identity with the team, become committed to the team and develop loyalty towards the remainder of the team. It avoided the necessity for the team to make assumptions in order to progress, which occurred in C1 due to the lack of participation by the customer. Continuous customer participation can be achieved by assigning a single customer to the team, enforcing their attendance at the SPM, recommending that the customer is co-located with the core development team, or where this is not feasible conducting regular site visits to the location of the development team.

R5. Use a tool to clearly define and document the short-term goals for each sprint

Each of the three teams studied defined and agreed on the common short-term goal for the sprint during the SPM, which was established collaboratively between the customer, where present, and the remainder of the team. The goals for each sprint were documented in the Sprint Backlog, and in the case of C2, they were also displayed on the Scrum Board. These were always visible or accessible to all team members. Consequently, there was no uncertainty over the goals for the sprint with team members committed to delivering the agreed functionality within the timeframe of the sprint.

R6. Assign project teams to a single project for each sprint

Assigning teams to multiple projects caused a number of difficulties for team members in C2 and should be avoided. Team members were expected to complete tasks on three different projects, yet planning in the SPM related to activities for one project. The tasks for the remaining two projects were documented outside of the SPM. Team members found it difficult and frustrating when planning in the SPM as all team members were not available to the project 100% of the time, which reduced

the amount of functionality that could be delivered to the customer. On occasion a team member may not be available to the project team for an entire sprint. Team members were required to state their availability for each project at the start of the SPM. It relied on the honesty of team members to indicate their availability for the sprint, which may not accurately reflect their actual availability. However, if teams are required to work on multiple projects at the same time, all activities should be incorporated into the SPM to provide an accurate picture of the effort required from each team member.

R7. Encourage team collaboration in all aspects of sprint planning

Team collaboration in the SPM helped team members to develop a sense of identity with the team and it also ensured that they became familiar with the norms and values of the team. While collaboration occurred to a certain degree in C1, it was not as prevalent as in C2 or C3. In C2 and C2 all tasks were collaborative from prioritising tasks in conjunction with the customer, estimating tasks, assigning tasks to personnel, and rotating tasks amongst team members. In C1 the Project Manager assigned tasks to team members outside of the meeting, who then determined estimates for their assigned tasks, which were verbally agreed by the team at the SPM. There was little disagreement on estimates or the assignment of tasks in this team at the SPM. This action had resulted in the creation of experts in particular areas as the Project Manager assigned tasks to the person most appropriate for the task.

R8. Ensure team members declare their availability for a sprint

The goal for the sprint must be clearly defined in conjunction with the customer with the team considering the resources available to them for the duration of the sprint, the time available, and the priority of tasks. It is important that team members honestly declare their availability for the sprint, which is made visible to all. This ensures that the team only commit to what is feasible to achieve within the timeframe of the sprint given the resources available. Where tasks are difficult to estimate sufficient time must be allocated in the sprint to investigate the task before committing to a final estimate. A commitment by the team to deliver additional

functionality can lead to overtime or the failure to deliver all the functionality agreed for the sprint.

R9. Avoid the necessity to continuously assign each task to two or more members

In the SPM teams have autonomy to make various decisions, which should be monitored to avoid the emergence of poor practices. For example, in C2 team members decided to allocate two team members to a number of tasks where the tasks assigned were complex. While this increased learning amongst team members it evolved into the assignment of two or more team members to each task, regardless of the complexity of the task, which reduced the productivity of the team. Practices such as this must be monitored carefully by the team and only implemented when absolutely necessary.

R10. Use the SPM to develop shared values and beliefs

In each of the three teams the SPM facilitated the development of several shared values and beliefs, such as cooperativeness, learning, communication, honesty, and accountability. Each of these can be developed in the SPM by ensuring continued participation of team members in the SPM, asking each team member to contribute, rotating tasks, sharing knowledge, allowing the team to work collaboratively, and giving them the freedom to decide how they will achieve their goals.

R11. Observe rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members

Each team implemented the SPM in the manner that suited their needs. This resulted in the development of various rules, regulations, and behaviours within the team to which the team members were expected to comply. Team members were allowed to sanction each other within reason for non-conformance to accepted behaviours, rules or regulations. However, this must be monitored carefully to ensure that team members do not become overly critical, or use non-conformance as an excuse to regularly criticise others.

R12. Avoid criticism or sanctioning of team members

In the SPM evaluation of team members and the corresponding reward or sanctioning of each other was limited. In one of the three teams studied evaluation and sanctioning related to the lack of conformance to rules, regulations, or behaviours. However, teams must ensure that evaluation does not become excessive or extreme and that the SPM does not provide a forum for team members to regularly criticise team members or provide negative feedback. Instead, constructive feedback should be provided, to allow team members to improve.

Resultant Characteristic of Clan Control	Recommendations
Develop a sense of identity	<p>R1: Ensure continued participation and contribution of all team members in the SPM and allow everyone to speak freely (C2, C3)</p> <p>R2: Ensure participation of new and distributed team members in the SPM from the outset and allow everyone to speak freely e.g. provide introductory activity for new team members, conduct site visits, use of video-conferencing (C1, C2)</p> <p>R3. Facilitate QA participation in the SPM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team (C1, C3)</p> <p>R4. Encourage and facilitate customer participation in the SPM to build relationships with the customer, to ensure their goals are aligned with those of the team, prioritisations are correct, avoiding the necessity for the team to make assumptions in order to progress (C2, C3)</p> <p>R5. Use a tool to clearly define and document the short-term goals for each sprint e.g. Sprint Backlog, Scrum Board that provides a single, easily accessible and transparent interface for team members (C1, C2, C3)</p> <p>R7. Encourage team collaboration in all aspects of sprint planning e.g. assigning tasks, prioritising tasks, task estimating, rotating tasks, which assists with learning and reduces the possibility of the development of experts in particular areas, who may make the decisions (C2, C3)</p>
Develop a sense of commitment	<p>R4. Encourage and facilitate customer participation in the SPM to build relationships with the customer, to ensure their goals are aligned with those of the team, prioritisations are correct, avoiding the necessity for the team to make assumptions in order to progress (C2, C3)</p> <p>R5. Use a tool to clearly define and document the short-term goals for each sprint e.g. Sprint Backlog, Scrum Board that provides a single, easily accessible and transparent interface for team members (C1, C2, C3)</p> <p>R8. Ensure team members declare their availability for a sprint e.g. ensure availability is visible to all, avoid overtime (C3)</p>
Develop a sense of loyalty	<p>R1: Ensure continued participation and contribution of all team members in the SPM and allow everyone to speak freely (C2, C3)</p> <p>R2: Ensure participation of new and distributed team members in the SPM from the outset and allow everyone to speak freely (C1, C2)</p> <p>R3. Facilitate QA participation in the SPM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team (C1, C3)</p> <p>R4. Encourage and facilitate customer participation in the SPM to build relationships with the customer, to ensure their goals are aligned with those of the team, prioritisations are correct, avoiding the necessity for the team to make assumptions in order to progress (C2, C3)</p>

Resultant Characteristic of Clan Control	Recommendations
Members develop common interests and goals	<p>R4. Encourage and facilitate customer participation in the SPM to build relationships with the customer, to ensure their goals are aligned with those of the team, prioritisations are correct, avoiding the necessity for the team to make assumptions in order to progress (C2, C3)</p> <p>R5. Use a tool to clearly define and document the short-term goals for each sprint e.g. Sprint Backlog, Scrum Board that provides a single, easily accessible and transparent interface for team members (C1, C2, C3)</p> <p>R6. Assign project teams to a single project for each sprint (C1, C3)</p> <p>R9. Avoid the necessity to continuously assign each task to two or more members (C1, C3)</p>
Rituals and ceremonies identify and reinforce acceptable behaviour	<p>R1: Ensure continued participation and contribution of all team members in the SPM and allow everyone to speak freely e.g. to reinforce acceptable behaviours (C2, C3)</p> <p>R2: Ensure participation of new and distributed team members in the SPM from the outset and allow everyone to speak freely e.g. to reinforce acceptable behaviours (C1, C2)</p> <p>R11. Observe rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members (C1, C2, C3)</p>
Members influenced by shared norms, values and beliefs	<p>R1: Ensure continued participation and contribution of all team members in the SPM and allow everyone to speak freely (C2, C3)</p> <p>R2: Ensure participation of new and distributed team members in the SPM from the outset and allow everyone to speak freely (C1, C2)</p> <p>R3. Facilitate QA participation in the SPM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team (C1, C3)</p> <p>R4. Encourage and facilitate customer participation in the SPM to build relationships with the customer, to ensure their goals are aligned with those of the team, prioritisations are correct, avoiding the necessity for the team to make assumptions in order to progress (C2, C3)</p> <p>R7. Encourage team collaboration in all aspects of sprint planning e.g. assigning tasks, prioritising tasks, task estimating, rotating tasks, which assists with learning and reduces the possibility of the development of experts in particular areas, who may make the decisions (C2, C3)</p> <p>R9. Avoid the necessity to continuously assign each task to two or more members (C3)</p> <p>R10. Use the SPM to develop shared values and beliefs e.g. communication, collaboration, honesty</p>
Members evaluate and reward or sanction each other	<p>R12. Avoid criticism or sanctioning of team members (C1, C3)</p>

Table 6.1-1 Recommendations for the Sprint Planning Meeting

A summary of the recommendations are presented in a different way, so a reader can quickly identify which characteristics of clan control can be achieved by the implementation of each recommendation.

Recommendation	Resultant Characteristic of Clan Control
R1 Encourage continued participation and contribution of all team members in the SPM and allow everyone to speak freely.	Develop a sense of identity Develop a sense of loyalty Rituals and ceremonies identify and reinforce acceptable behaviour Members influenced by shared norms, values and beliefs
R2 Encourage participation of new and distributed team members in the SPM from the outset and allow everyone to speak freely e.g. provide introductory activity for new team members, conduct site visits, use of video-conferencing	Develop a sense of identity Develop a sense of loyalty Rituals and ceremonies identify and reinforce acceptable behaviour Members influenced by shared norms, values and beliefs
R3 Facilitate QA participation in the SPM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team	Develop a sense of identity Develop a sense of loyalty Members influenced by shared norms, values and beliefs
R4 Encourage and facilitate customer participation in the SPM e.g. to build relationships with the customer, to ensure their goals are aligned with those of the team, prioritisations are correct, and avoiding the necessity for the team to make assumptions in order to progress	Develop a sense of identity Develop a sense of commitment Develop a sense of loyalty Members develop common interests and goals Members influenced by shared norms, values and beliefs
R5 Use a tool to clearly define and document the short-term goals for each sprint e.g. Sprint Backlog, Scrum Board that provides a single, easily accessible and transparent interface for team members	Develop a sense of identity Develop a sense of commitment Members develop common interests and goals
R6 Assign project teams to a single project for each sprint	Members develop common interests and goals
R7 Encourage team collaboration in all aspects of sprint planning .g. assigning tasks, prioritising tasks, task estimating, rotating tasks, avoid the development of experts in particular areas	Develop a sense of identity Members influenced by shared norms, values and beliefs
R8 Ensure team members declare their availability for a sprint	Develop a sense of commitment
R9 Avoid the necessity to continuously assign each task to two or more members	Members develop common interests and goals Members influenced by shared norms, values and beliefs
R10 Use the SPM to develop shared values and beliefs e.g. communication, collaboration, honesty	Members influenced by shared norms, values and beliefs
R11 Observe rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members	Rituals and ceremonies identify and reinforce acceptable behaviour
R12 Avoid criticism or sanctioning of team members	Members evaluate and reward or sanction each other

Table 6.1-2 Recommendations for the Sprint Planning Meeting summarised by characteristic

6.2 Daily Scrum Meeting Recommendations

As was seen in Table 5.4-1 (p.238) the DSM helped to achieve 6 of the 7 resultant characteristics in each of the three cases either wholly or partially, which indicated that the DSM contributed to the development of clan control. Each of the cases studied had implemented the DSM in different ways. Some practices were implemented well, whereas others were poorly executed by the teams, or had possibly developed over time. Both good and poor practices are discussed with recommendations made where the practice was strong or weak in particular teams (Table 6.2-1). The recommendations made for the weak practices were based on good practices in other teams. These are indicated in the table by providing the label for the case(s) after each recommendation. Each recommendation is labelled with the letter ‘R’ followed by a number in increasing order. Note that the same recommendation may correspond to a number of different resultant characteristics of clan control.

R1: Ensure continued participation and contribution of all team members in the DSM and allow everyone to speak freely

Similar to the SPM (section 6.1) where team members regularly participated in and contributed to the DSM it helped them to develop their identity within the team, develop a sense of loyalty to the team, and allowed them to be influenced by the shared norms, values and beliefs that existed within the team, resulting in compliance with acceptable behaviours as defined by the team. Teams in C2 and C3 adhered to this good practice and this was evident in both of these teams. Reasons relating to how this good practice was implemented were detailed in section 6.1 (p.239).

R2: Ensure participation of new and distributed team members in the DSM from the outset and allow everyone to speak freely

The introduction of new and distributed team members to the DSM from the outset proved valuable for these individuals. It quickly integrated them into the team where they became familiar with team members, the norms of the team, the behaviours considered acceptable by the team, the nature of the project, the work that was under development, and ensured that they communicated and interacted with the team

immediately, even if they had little to say at the start. It also ensured that the goals of all team members were aligned. All of this helped to develop their sense of identity within the team and develop their sense of loyalty to the team. This practice was enforced in both C1 and C2 where new and distributed team members were introduced to the team. In C2 the team made a concerted effort to integrate a new team member in the DSM by providing an introductory activity (unrelated to the project) in the SPM for the new team member to complete. This team member was required to complete the sub-tasks defined at the SPM and report on their progress in the DSM. It allowed the new team members to participate in the ritual of the DSM and become quickly familiar with the norms of the practice. C1 had distributed team members across three different continents, which was not conducive to daily communication between all distributed team members due to the time differences. This resulted in two separate DSM's – one conducted between the Project Manager and the team, based in India and a separate DSM between the Irish and American teams. This likely impacted on the relationships between the team based in India and the remaining team members, their loyalty to each other and their sense of identity within the team as they did not benefit from daily communication and interaction with all team members.

R3. Facilitate QA participation in the DSM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team

QA were regularly present in the DSM in C3, which helped to develop a loyalty between them and the remainder of the team that was not present previously. It also helped QA to develop their own identity within the team, and allowed them to influence the shared norms, values and beliefs that existed within the team by vocalising the needs of QA and ensuring that team members were aware of how to improve their software delivered from a quality perspective. This good practice was not prevalent in C1, which was due to the distributed nature of the QA team in C1, the frequent change in QA personnel, as well as the cultural differences between the QA team and the development team. The time difference between the distributed team members (USA, Ireland and India) was not conducive to daily communication between all distributed team members. This resulted in the Project Manager conducting a DSM between himself and the QA team in the morning followed by a

separate DSM between the Irish and American teams in the afternoon. The lack of daily interaction and communication between the QA team and the other team members impacted on their ability to develop relationships with each other, their loyalty to each other and their sense of identity within the team. As the core development team was based in Ireland the DSM could rotate on a sprint basis where one sprint the DSM is held regularly between the Irish and the American teams, and the following sprint it is held regularly between the Irish and Indian teams in order to improve links between team members and avoid continually disadvantaging the Indian team.

R4. Encourage and facilitate customer participation in the DSM

Customer representation was present in C2 and C3 in the form of the Product Owner, who was co-located with the team. Active participation in the DSM by the customer helped the customer to develop a sense of identity with the team, become committed to the team and develop loyalty towards the remainder of the team. They were available on a daily basis to provide continuous advice and feedback to the team. It avoided the necessity for the team to make assumptions in order to progress, which occurred in C1 due to the lack of participation by the customer. Continuous customer participation can be achieved by assigning a single customer to the team, enforcing their attendance at the DSM, recommending that the customer is co-located with the core development team, or where this is not feasible conducting regular site visits to the location of the development team.

R5. Ensure daily tasks and deliverables are clearly visible, accessible, and transparent to all team members

Daily goals for team members were discussed at the DSM. Providing teams with visibility and transparency on the goals of individual team members e.g. the person assigned to each task, the associated estimate for each task, progress made to date, helped team members to feel personally accountable for their tasks, gave them ownership over their tasks and contributed to the development of their sense of identity within the team, their personal commitment to the team, and their loyalty to the team. It also allowed team members to monitor the progress of others to ensure that team members did not deviate from their assigned tasks, or were not exceeding

their estimated time. Monitoring of tasks was achieved by the use of a Scrum Board in C2, which was displayed on the wall, and the use of an electronic Scrum Board in C1 and C3 that was accessible to team members at any time.

R6. Encourage open and honest discussions that also facilitates requests for assistance

All three teams had created a very supportive environment at the DSM that welcomed the contribution and opinions of all team members. Team members did not criticise each other or their opinions. Instead discussions were open and frank with input provided by those who had a view or a suggested solution to a particular problem, which were listened to by all. Team members were encouraged to request assistance and provide feedback and it was expected that a team member would request help if it was needed. A team member was more likely to be reprimanded for not seeking assistance, rather than reporting a lack of progress.

R7. Allow the team to renegotiate deliverables for the sprint

The DSM provided the team with the ability to renegotiate deliverables for the sprint. While this was not ideal, situations did arise in each of the three teams where this was necessary and the DSM provided the teams with the flexibility to do this. Occasionally teams underestimated the amount of effort required for specific tasks at the SPM. Instead of failing to complete all tasks for the sprint the team they had the ability to renegotiate with the customer to move some of the lower priority tasks to a subsequent sprint. This ensured that the team delivered completed functionality to the customer. On other occasions the customer requested additional functionality mid-sprint, which required the customer to re-prioritise tasks and negotiate functionality with the team.

R8. Observe rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members

Each team implemented the DSM in the manner that suited their needs. This resulted in the development of various rules, regulations, and behaviours within the team to which the team members were expected to comply. Even though behaviours may be determined as acceptable by the team, they may not be in the best interests of the team. For example, in C2 and C3 behaviour had emerged within the team that it was

acceptable for team members to fail to update their estimate for a task on the Scrum Board, if the amount of time required increased. This resulted in an inaccurate portrayal of progress towards the common goal. Such behaviour must be identified, which in C2 was identified by the Product Owner and the Scrum Master, and highlighted to the team, so that they can correct this behaviour. In the DSM team members were also allowed to sanction each other within reason for non-conformance to accepted behaviours, rules or regulations. However, this must be monitored carefully to ensure that team members do not become overly critical, or use non-conformance as an excuse to regularly criticise others.

R9. Use the daily scrum to evaluate the progress of team members

The DSM was used in the three teams to monitor and evaluate the daily progress of team members and ensured they were working towards the common goal. This was achieved in C2 by focusing on and referring to the Scrum Board. In C1 and C3 the Scrum Board was not used at the meeting, but team members updated the progress of their tasks on the Scrum Board daily, which could be accessed by any team member at any time. To some extent team members in C1 and C3 monitored the verbal reports of progress of specific tasks, particularly where there were dependencies between tasks and could identify where team members had difficulty and required assistance. In both of these cases the scrum board was not visible during the meeting.

R10. Proactively use the daily scrum to verbally reward and praise members

The DSM was used in C2 to verbally reward and praise team members on completion of a successful task, which was welcomed by team members. While it is not necessary to praise team members on the successful completion of each task, it may be beneficial to provide some praise or recognition when a difficult or complex task is finished. There was no evidence of this in C1 or C3.

R11. Avoid sanctioning team members

The DSM could become a forum for sanctioning or criticising team members on a daily basis, which could develop negative attitudes within the team. However, in each of the three cases studied sanctioning was minimal and where there was

criticism it was constructive and for the betterment of the team. The only evidence of sanctioning related to the non-conformance to the norms and behaviours of the team.

Resultant Characteristic of Clan Control	Recommendations
Develop a sense of identity	<p>R1. Ensure continued participation and contribution of all team members in the DSM and allow everyone to speak freely (C2, C3)</p> <p>R2. Ensure participation of new and distributed team members in the DSM from the outset and allow everyone to speak freely (C1, C2)</p> <p>R3. Facilitate QA participation in the DSM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team (C1, C3)</p> <p>R4. Encourage and facilitate customer participation in the DSM to ensure they are available to provide advice and feedback to the team (C2, C3)</p> <p>R5. Ensure daily tasks and deliverables are clearly visible, accessible, and transparent to all team members e.g. Scrum Board (C1, C2, C3)</p> <p>R6. Encourage open and honest discussions that also facilitates requests for assistance (C1, C2, C3)</p>
Develop a sense of commitment	<p>R2. Ensure participation of new and distributed team members in the DSM from the outset and allow everyone to speak freely (C1, C2)</p> <p>R4. Encourage and facilitate customer participation in the DSM to ensure they are available to provide advice and feedback to the team (C2, C3)</p> <p>R5. Ensure daily tasks and deliverables are clearly visible, accessible, and transparent to all team members e.g. Scrum Board (C1, C2, C3)</p> <p>R6. Encourage open and honest discussions that also facilitates requests for assistance (C1, C2, C3)</p>
Develop a sense of loyalty	<p>R1. Ensure continued participation and contribution of all team members in the DSM and allow everyone to speak freely (C2, C3)</p> <p>R2. Ensure participation of new and distributed team members in the DSM from the outset and allow everyone to speak freely (C1, C2)</p> <p>R3. Facilitate QA participation in the SPM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team (C1, C3)</p> <p>R4. Encourage and facilitate customer participation in the DSM to ensure they are available to provide advice and feedback to the team (C2, C3)</p> <p>R5. Ensure daily tasks and deliverables are clearly visible, accessible, and transparent to all team members e.g. Scrum Board (C1, C2, C3)</p> <p>R6. Encourage open and honest discussions that also facilitates requests for assistance (C1, C2, C3)</p>
Members develop common interests and goals	<p>R2. Ensure participation of new and distributed team members in the DSM from the outset and allow everyone to speak freely (C1, C2)</p> <p>R4. Encourage and facilitate customer participation in the DSM to ensure they are available to provide advice and feedback to the team (C2, C3)</p> <p>R5. Ensure daily tasks and deliverables are clearly visible and transparent to all team members e.g. Scrum Board (C1, C2, C3)</p> <p>R7. Allow the team to renegotiate deliverables for the sprint (C1, C2, C3)</p>
Rituals and ceremonies identify and reinforce acceptable behaviour	<p>R1. Ensure continued participation and contribution of all team members in the DSM and allow everyone to speak freely (C2, C3)</p> <p>R2. Ensure participation of new and distributed team members in the DSM from the outset and allow everyone to speak freely e.g. to reinforce acceptable behaviours (C1, C2)</p> <p>R8. Observe rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members (C1, C2, C3)</p>

Resultant Characteristic of Clan Control	Recommendations
Members influenced by shared norms, values and beliefs	R1. Ensure continued participation and contribution of all team members in the DSM and allow everyone to speak freely (C2, C3) R2. Ensure participation of new and distributed team members in the DSM from the outset and allow everyone to speak freely (C1, C2) R5. Ensure daily tasks and deliverables are clearly visible, accessible, and transparent to all team members e.g. Scrum Board (C1, C2, C3) R6. Encourage open and honest discussions that also facilitates requests for assistance (C1, C2, C3)
Members evaluate and reward/sanction each other	R9. Use the daily scrum to evaluate the progress of team members (C1, C2, C3) R10. Proactively use the DSM to verbally reward and praise team members (C2) R11. Avoid sanctioning team members (C1, C2, C3)

Table 6.2-1 Recommendations for the Daily Scrum Meeting

A summary of the recommendations are presented in a different way, so a reader can quickly identify which characteristics of clan control can be achieved by the implementation of each recommendation.

Recommendation	Resultant Characteristic of Clan Control
R1 Ensure continued participation and contribution of all team members in the DSM and allow everyone to speak freely	Develop a sense of identity Develop a sense of loyalty Rituals and ceremonies identify and reinforce acceptable behaviour Members influenced by shared norms, values and beliefs
R2 Ensure participation of new and distributed team members in the DSM from the outset and allow everyone to speak freely	Develop a sense of identity Develop a sense of commitment Develop a sense of loyalty Members develop common interests and goals Rituals and ceremonies identify and reinforce acceptable behaviour Members influenced by shared norms, values and beliefs
R3 Facilitate QA participation in the DSM to ensure their needs are accounted for and to build relationships between QA and the remainder of the team	Develop a sense of identity Develop a sense of loyalty
R4 Encourage and facilitate customer participation in the DSM	Develop a sense of identity Develop a sense of commitment Develop a sense of loyalty Members develop common interests and goals
R5 Ensure daily tasks and deliverables are clearly visible, accessible, and transparent to all team members	Develop a sense of identity Develop a sense of commitment Develop a sense of loyalty Members develop common interests and goals Members influenced by shared norms, values and beliefs
R6 Encourage open and honest discussions that also facilitates requests for assistance	Develop a sense of identity Develop a sense of commitment Develop a sense of loyalty Members influenced by shared norms, values and beliefs
R7 Allow the team to renegotiate deliverables for the sprint	Members develop common interests and goals
R8 Observe rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members	Rituals and ceremonies identify and reinforce acceptable behaviour
R9 Use the daily scrum to evaluate the progress of team members	Members evaluate and reward/sanction each other
R10 Proactively use the daily scrum to verbally reward and praise team members	Members evaluate and reward/sanction each other
R11 Avoid sanctioning team members	Members evaluate and reward/sanction each other

Table 6.2-2 Recommendations for the Daily Scrum Meeting summarised by characteristic

6.3 Sprint Review and Retrospective Meeting Recommendations

As was shown in Table 5.4-1, p.238 the SRRM helped to achieve 6 of the 7 resultant characteristics in each of the three cases either wholly or partially, which indicated that the SRRM contributed to the development of clan control. Each of the cases studied had implemented the SRRM in different ways. Some practices were implemented well, whereas others were poorly executed by the teams, or had possibly developed over time. Both good and poor practices are discussed with recommendations made where the practice was strong or weak in particular teams (Table 6.3-1). The recommendations made for the weak practices were based on good practices in other teams. These are indicated in the table by providing the label for the case(s) after each recommendation. Each recommendation is labelled with the letter ‘R’ followed by a number in increasing order. Note that the same recommendation may correspond to a number of different resultant characteristics of clan control.

R1: Ensure continued participation and contribution of all team members in the SRRM and allow everyone to speak freely

Similar to the SPM and the DSM (section 6.1 and section 6.2) where team members regularly participated in and contributed to the SRRM it helped them to develop their identity within the team, develop a sense of loyalty to the team, and allowed them to be influenced by the shared norms, values and beliefs that existed within the team, resulting in compliance with acceptable behaviours as defined by the team. Teams in C2 and C3 adhered to this good practice and this was evident in both of these teams. Reasons relating to how this good practice was implemented are detailed in section 6.1 (p.239).

R2: Ensure participation of new and distributed team members in the SRRM from the outset and allow everyone to speak freely

The introduction of new and distributed team members to the SRRM from the outset was valuable for these individuals. It quickly integrated them into the team where they became familiar with team members, the norms of the team, the behaviours considered acceptable by the team, the nature of the project, the work that was under development, and ensured that they communicated and interacted with the team

immediately, even if they had little to say at the start. All of this helped to develop their sense of identity within the team and develop their sense of loyalty to the team. This practice was enforced in C1 where new and distributed team members were introduced to the team. This was not yet enforced in C2 as a new team member had not joined the team since the implementation of Scrum. However, it is reasonable to assume that the newest team member will participate in the SRRM at the end of his first sprint, which has yet to take place. As discussed in section 6.1 (p.239) some efforts were made in C1 to conduct site visits between Ireland and the USA to assist with relationship building between the two teams and the integration of distributed team members into the team. However, no site visits were reported between team members in India and the other two distributed sites. Communication and interaction between team members could also be improved through the use of technologies such as video-conferencing, which were available for use within the organisation.

R3. Facilitate QA participation in the SRRM to ensure they have an opportunity to provide feedback to the team

Similar to the SPM, QA were regularly present in the SRRM in both C1 and C3, which helped to develop a loyalty between them and the remainder of the team. It also helped QA to develop their identity within the team, and allowed them to influence the shared norms, values and beliefs that existed within the team by providing feedback to the team from a QA perspective and ensuring that team members were aware of how to improve the quality of the software delivered in future sprints. This good practice was more prevalent in C3 than in C1, which was partially due to the distributed nature of the QA team in C1, the frequent change in QA personnel, as well as the cultural differences between the QA team and the development team. The QA team in C1 were less vocal than the team in C3 when communicating feedback and their needs to the remainder of the team. The cultural difficulties with QA in C1 could be addressed to some extent by site visits between the two countries. However, this team are resolving the difficulties with QA by re-locating the QA function to Ireland where they will work alongside the core software development team.

R4. Encourage and facilitate customer participation in the SRRM

Customer representation was present in C2 and C3 in the form of the Product Owner, who was co-located with the team. Active participation in the SRRM by the customer helped the customer to develop a sense of identity with the team, become committed to the team and develop loyalty towards the remainder of the team. The customer had the opportunity at the end of each sprint to review the completed software and ensure that it had achieved the goals defined at the SPM. The customer also had an opportunity to provide feedback to the team, which could be taken into consideration in future sprints. Continuous customer participation can be achieved by assigning a single customer to the team, enforcing their attendance at the SRRM, recommending that the customer is co-located with the core development team, or where this is not feasible by conducting regular site visits to the location of the development team to help build relationships.

R5. Ensure team members individually reflect on work completed and proactively identify areas for improvement

The SRRM was implemented effectively in both C2 and C3 with each team member requested in turn to reflect personally on the work they completed and identify areas for improvement. In C1 team members recounted what work they had completed in the sprint, rather than reflecting on how they had completed it. There was no requirement to identify areas for improvement, which should be enforced by the Project Manager, who chaired the meeting. Where recommendations for improvement are suggested actions must be identified and tasks assigned to team members to address, where appropriate. Where actions are taken as a consequence of a recommendation by a particular team member it helps to increase a sense of identity, loyalty and commitment to the team. It also helps to develop a common goal within the team to improve, identifies and reinforces good behaviour within the team, and influences other team members to engage in the process and identify areas for improvement.

R6. Observe rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members

The implementation of the SRRM resulted in the development of various rules, regulations, and behaviours within each team to which the team members were expected to comply. Even though behaviours may be determined as acceptable by the team, they may not be in the best interests of the team. For example, in C1 there was limited engagement or discussion at the meeting, with little reflection on tasks completed and no areas for improvement identified. However, the Project Manager acknowledged the poor use of the practice and has plans to improve the practice, and encourage further discussion, to ensure the team receive benefit from the practice.

R7. Use the SRRM to evaluate adherence to the goal defined in the SPM

All three teams used the SRRM to evaluate their compliance with and fulfilment of the goals defined in the SPM, which was good practice. Where the teams did not fulfil their defined goals they could move tasks to subsequent sprints.

R8. Proactively use the SRRM to verbally reward and praise team members

The SRRM was proactively used in C2 and C3 to verbally provide praise and reward for work completed. This was provided by team members to each other, by the customer and in C3 by senior management. There was no evidence that this occurred in C1, which could be undertaken by the Project Manager in the future where particularly difficult tasks or sprints were completed. The Project Manager could also use the SRRM as a means by which positive feedback is reported from the customer, to highlight where a particular individual may have made an outstanding contribution to the team during the sprint.

R9. Avoid sanctioning team members

There was no evidence in any of the three teams that the SRRM was used to sanction team members. However, in other teams it could be used as a forum to sanctioning or criticise team members which could develop negative attitudes within the team. It is recommended to avoid this, or if it is necessary that criticism or sanctioning of team members is constructive.

Resultant Characteristic of Clan Control	Recommendations
Develop a sense of identity	<p>R1. Ensure continued participation and contribution of all team members in the SRRM and allow everyone to speak freely (C2, C3)</p> <p>R2. Ensure participation of new and distributed team members in the SRRM from the outset and allow everyone to speak freely (C1, C2)</p> <p>R3. Facilitate QA participation to ensure they have the opportunity to provide feedback to the team from a quality perspective (C3)</p> <p>R4. Encourage and facilitate customer participation to ensure they have the opportunity to review the software delivered, provide feedback to the team, and determine that the team have delivered the functionality as agreed at the SPM, and avoid a blame culture (C2, C3)</p> <p>R5. Ensure team members individually reflect on work completed and proactively identify areas for improvement (C2, C3)</p>
Develop a sense of commitment	<p>R4. Encourage and facilitate customer participation to ensure they have the opportunity to review the software delivered, provide feedback to the team, and determine that the team have delivered the functionality as agreed at the SPM, and avoid a blame culture (C2, C3)</p> <p>R5. Ensure team members individually reflect on work completed and proactively identify areas for improvement (C2, C3)</p>
Develop a sense of loyalty	<p>R1. Ensure continued participation and contribution of all team members in the SRRM and allow everyone to speak freely (C2, C3)</p> <p>R2. Ensure participation of new and distributed team members in the SRRM from the outset and allow everyone to speak freely (C1, C2)</p> <p>R3. Facilitate QA participation to ensure they have the opportunity to provide feedback to the team from a quality perspective (C3)</p> <p>R4. Encourage and facilitate customer participation to ensure they have the opportunity to review the software delivered, provide feedback to the team, and determine that the team have delivered the functionality as agreed at the SPM, and avoid a blame culture (C2, C3)</p> <p>R5. Ensure team members individually reflect on work completed and proactively identify areas for improvement (C2, C3)</p>
Members develop common interests and goals	<p>R5. Ensure team members individually reflect on work completed and proactively identify areas for improvement (C2, C3)</p>
Rituals and ceremonies identify and reinforce acceptable behaviour	<p>R1. Ensure continued participation and contribution of all team members in the SRRM and allow everyone to speak freely e.g. to reinforce acceptable behaviours where everyone can speak freely (C2, C3)</p> <p>R2. Ensure participation of new and distributed team members in the SRRM from the outset and allow everyone to speak freely e.g. to reinforce acceptable behaviours where everyone can speak freely (C1, C2)</p> <p>R4. Encourage and facilitate customer participation to ensure they have the opportunity to review the software delivered, provide feedback to the team, and determine that the team have delivered the functionality as agreed at the SPM (C2, C3)</p> <p>R5. Ensure team members individually reflect on work completed and proactively identify areas for improvement (C2, C3)</p> <p>R6. Monitor rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members (C1, C2, C3)</p>

Resultant Characteristic of Clan Control	Recommendations
Members influenced by shared norms, values and beliefs	R1. Ensure continued participation and contribution of all team members in the SRRM and allow everyone to speak freely (C2, C3) R2. Ensure participation of new and distributed team members in the SRRM from the outset and allow everyone to speak freely (C1, C2) R3. Facilitate QA participation to ensure they have the opportunity to provide feedback to the team from a quality perspective and can highlight areas for improvement from a quality perspective, which the team take on board in future sprints (C3) R5. Ensure team members individually reflect on work completed and proactively identify areas for improvement (C2, C3)
Members evaluate and reward/sanction each other	R7. Use the SRRM to evaluate adherence to the goal defined in the SPM (C1, C2, C3) R8. Proactively use the SRRM to verbally reward and praise team members (C2, C3) R9. Minimise sanctioning team members (C1, C2, C3)

Table 6.3-1 Recommendations for the Sprint Review and Retrospective Meeting

A summary of the recommendations are presented in a different way, so a reader can quickly identify which characteristics of clan control can be achieved by the implementation of each recommendation.

SRRM Recommendation	Resultant Characteristic of Clan Control	
R1	Ensure continued participation and contribution of all team members in the SRRM.	Develop a sense of identity Develop a sense of loyalty Rituals and ceremonies identify and reinforce acceptable behaviour Members influenced by shared norms, values and beliefs
R2	Ensure participation of new and distributed team members in the SRRM from the outset e.g. to reinforce acceptable behaviours	Develop a sense of identity Develop a sense of loyalty Rituals and ceremonies identify and reinforce acceptable behaviour Members influenced by shared norms, values and beliefs
R3	Facilitate QA participation in the SRRM to ensure they have an opportunity to provide feedback to the team from a quality perspective and can highlight areas for improvement from a quality perspective, which the team take on board in future sprints	Develop a sense of identity Develop a sense of loyalty Members influenced by shared norms, values and beliefs
R4	Encourage and facilitate customer participation to ensure they have the opportunity to review the software delivered, provide feedback to the team, and determine that the team have delivered the functionality as agreed at the SPM and avoid a blame culture	Develop a sense of identity Develop a sense of commitment Develop a sense of loyalty Rituals and ceremonies identify and reinforce acceptable behaviour
R5	Ensure team members individually reflect on work completed and proactively identify areas for improvement	Develop a sense of identity Develop a sense of commitment Develop a sense of loyalty Members develop common interests and goals Rituals and ceremonies identify and reinforce acceptable behaviour Members influenced by shared norms, values and beliefs
R6	Observe rules, regulations, and behaviours that develop to ensure they are reasonable and acceptable to all team members	Rituals and ceremonies identify and reinforce acceptable behaviour
R7	Use the SRRM to evaluate adherence to the goal defined in the SPM	Members evaluate and reward/sanction each other
R8	Proactively use the SRRM to verbally reward and praise team members	Members evaluate and reward/sanction each other
R9	Avoid sanctioning team members	Members evaluate and reward/sanction each other

Table 6.3-2 Recommendations for the Sprint Review and Retrospective Meeting summarised by characteristic

6.4 Summary

This chapter has presented recommendations on how three agile practices may be improved in the teams studied to achieve higher levels of clan control. These are based on data gathered from three industry-based ASD teams and they provide an insight on how each team has tailored the practice to suit their unique environment, which differs from the textbook version of the practice. This is similar to the tailoring and implementation of software development methods, otherwise known as the ‘method-in-action’ (Fitzgerald et al., 2002). It reveals where each of the teams exhibited good practices and where poor practices were used, that negatively impacted on the teams, such as the productivity of the team, or criticism of team members. Teams should avoid these types of practices, or where present, the team should identify and address the difficulties that they may cause.

Some of these recommendations are not novel and advocate the use of the practice as was intended by the prescribed methodology that is detailed in textbooks. However, there were some interesting effects and implementations of the practices which contributed to clan control in the team such as the acceptance of a lack of participation in the SRRM (C1), the assignment of two individuals to a single task (C2) and the active participation of QA (C3). This demonstrates that teams should only use the textbook version of the methodology as a starting point and then tailor the methodology and its practices to fit the environment (DeMarco, 1982, p13). The next chapter examines the findings from this study in the context of the literature discussed in Chapter 2 and Chapter 3, and will detail their significance and implications for clan control in ASD teams.

7 Discussion and Conclusion

7.0 Introduction

This chapter summarises the main findings of this study. It is structured as follows. First, the research objective is revisited with detail on how each objective has been achieved in this study. The subsequent section discusses the findings of the study in the context of the literature detailed in Chapter 2 and Chapter 3 and the framework presented in Figure 3.5-1 (p.77) and highlights the significance and implication of agile practices for clan control in ASD teams. The theoretical and practical contribution of this study are then presented (section 7.3). Finally, limitations of the research are identified (section 7.4), recommendations for future research are documented (section 7.5), and the chapter concludes (section 7.6).

7.1 Review of Research Objective

This study investigated clan control in ASD teams, which was underpinned by control theory. An examination of the literature provided evidence of several control frameworks, which incorporated both formal and informal controls studied in different disciplines such as accounting, management, and marketing. These frameworks focused on control at a broad level rather than on a single specific control and divided control into two main categories: formal control and informal control. This study elected to focus on one single control, namely clan control, as it is one of the lesser studied modes of control. Clan control also focuses on people and their relationships, which are important aspects of team work, and are particularly important in ASD teams where collegiality and cooperation is paramount. While several studies detailed different aspects and elements of clan control a gap was identified in relation to the lack of a comprehensive clan control framework. This was the first objective that this study attempted to address by developing a framework that detailed the pre-conditions necessary for clans to develop and the pre-conditions required to implement clan control, which help to achieve the resultant characteristics of clan control.

The application of control theory in the context of ISD teams and subsequently ASD teams was then explored. There were few studies that applied control theory to ASD teams (Cram and Brohman, 2010; Harris et al., 2009; Maruping et al., 2009) with none identified that specifically focused on clan control in ASD teams. As an ASD team can be viewed as a clan and have substantial control over how they operate, informal controls such as clan control may be present and may be an appropriate form of control for such teams. ASD teams implement various agile practices to assist with the development of software with few studies investigating the impact of specific agile practices on ASD teams. Therefore, this study chose to investigate whether three agile practices, the sprint planning meeting, the daily scrum meeting, and the sprint review and retrospective meeting helped to achieve the resultant characteristics of clan control in ASD teams, which addressed the second research objective. The final research objective was aimed at practitioners with the intention of identifying how ASD teams could modify agile practices to help achieve greater levels of clan control. From a review of the literature published in the mainstream outlets, no such investigation has been conducted to date.

The overall research objectives to guide the study were as follows:

- (a) Develop a framework of the pre-conditions and resultant characteristics of clan control identified in the literature
- (b) Use this framework to investigate how agile practices help ASD teams to achieve the resultant characteristics of clan control
- (c) Identify how agile practices may be improved to achieve higher levels of clan control

The first research objective was addressed in Chapter 2 where the literature on control and control theory in an organisational context was reviewed and analysed to develop a framework for clan control. This framework details the conditions required to develop clans within an organisation, followed by the pre-conditions to implement clan control within organisations. When these pre-conditions exist they can help to achieve clan control and result in specific characteristics that demonstrate the existence of clan control.

The second research objective was answered in Chapter 5 using a multiple-case study approach. This chapter examined how three agile practices helped to achieve the resultant characteristics of clan control in ASD teams. The data gathered was analysed for each concept to determine if the agile practices helped to achieve a particular characteristic of clan control. The third research objective was answered in Chapter 6 where recommendations were developed on how each of the agile practices could be improved to help achieve clan control. These recommendations were based on good practices from the cases studied.

7.2 Discussion

This section discusses the findings from Chapter 5 in the context of the literature presented in Chapter 2 and Chapter 3 in an attempt to explain the findings summarised in Table 5.4-1 on p.238. The objective was to examine each of the resultant characteristics of clan control presented in the conceptual framework (Figure 7.2-1) and attempt to understand how each of the three agile practices helped to achieve them, which provides the structure for this discussion.

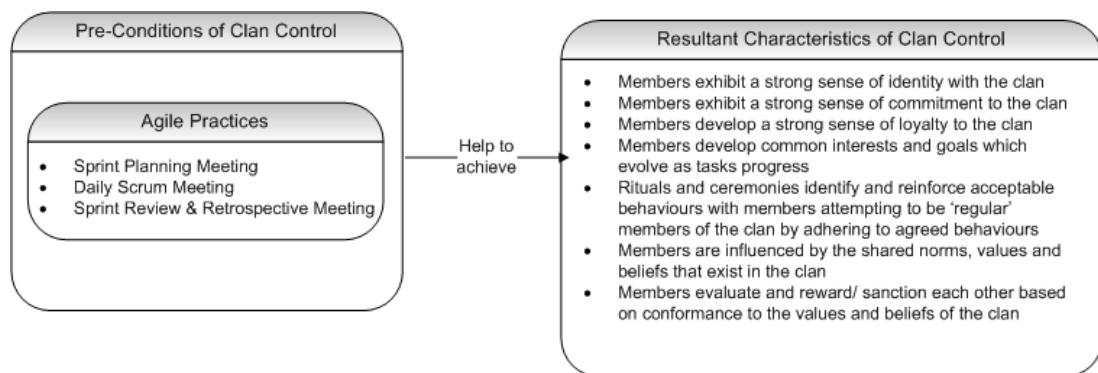


Figure 7.2-1 Conceptual framework used for this Study

7.2.1 Members exhibit a strong sense of identity with the clan

The three agile practices helped to achieve a strong sense of identity within the team in C2 and C3 whereas the evidence was mixed in C1. The findings suggest that the strong sense of identity in C2 and C3 could be attributed to the co-location of team members, their regular communication and interaction, and participation in and contribution to the three practices, which according to the literature helps to develop

a common identity (Whitworth and Biddle, 2007b). In both of these teams the customer regularly participated in the three practices and collaborated with the team, which helped to develop their sense of identity within the team, which is consistent with the literature (Asproni, 2004; Whitworth and Biddle, 2007b). These findings concur with the literature that regular interaction and communication can contribute to a sense of identity within a team.

The length of time an individual is employed by an organisation or are part of a team can help to increase their level of identity. In C2 several team members of the team were employed by the organisation for many years and had also worked together for a long period of time. Only two team members were employed elsewhere for a period of two years prior to joining this organisation. They were since employed by their current organisation for 8 and 10 years respectively. It is likely that the long-term employment of several individuals with the organisation and their lack of employment elsewhere contributed to their sense of identity with the team. This differed with C3 who also had a strong sense of identity with the team even though the individuals on this team were employed by their organisation for an average of 5 years with most having at least one other employer prior to joining this organisation. In addition, this team consisted of 6 different European nationalities, each of which have different cultures and traditions, yet team members had a strong sense of identity with their team. This suggests that regardless of the diversity of the team in terms of culture, experience, or length of service with an organisation, the three agile practices studied may help team members to develop a sense of identity with their team, but it may also be related to other factors that were not examined as part of this study.

The sense of identity with the team in C1 was weaker for a number of reasons. Agile methodologies were in use in this team for a period of two years and it is possible that interest in the agile practices had diminished over time as team members did not attend the meetings as regularly as they should or they limited their contribution at the meetings. Their lack of participation could be a contributing factor to a reduced sense of identity with the team. However, it could also indicate that that a low sense of identity with the team contributed to their lack of participation, although there was no evidence to suggest this. In addition, this team was distributed across three

continents and had to deal with the difficulties of multiple cultures and geographic distances. Seven of the team were Irish with two QA personnel based in India and one team member based in the United States. As the Indian team did not participate in the DSM with the remainder of the team due to differences in time zones, it was difficult for the Irish and American team members to build a continuous relationship with them or develop a common identity. This is recognised as a problem by Braithwaite and Joyce (2005) for distributed team members, even though in C1 QA were considered part of the team and communications did occur outside of the meeting. However, a lack of trust was evident between the Irish and Indian team with the Project Manager closely monitoring their work during the practices to ensure that they were honest in terms of the progress that was communicated. While C3 contained multiple cultures, they did not experience the same difficulty as C1, which likely related to the different cultures (European Vs, Asian). All team members in C3 were European, whereas in C1 the distributed team members were Indian, where difficulties with trust between Indian and western cultures have previously been identified (Nicholson and Sahay, 2001). This suggests that where team members are from similar cultures, such as a Western culture, developing a sense of identity with the team is less challenging. Further, the QA personnel in C1 changed several times over the two year period of the project, which resulted in a continuous need to develop a common identity with the new personnel. This implies that where there is regular turnover of staff it is difficult for team members to develop a sense of identity with each other. Yet, the single distributed team member who did participate in the study and was part of the team for 10 months did feel a sense of identity with the team, which he felt was contributed to by regular communication and interaction through the agile practices. This insinuates that the agile practices can help to alleviate communication barriers when distributed team members participate on a regular basis.

A further difficulty for team identity in C1 related to the lack of customer involvement in the agile practices. Customer involvement and cooperation with the rest of the team were identified as critical success factors for ASD teams (Chow and Cao, 2008; Misra et al., 2009; Nerur et al., 2005). The customer in C1, who was distributed, did not participate in the team and had little involvement in defining the

goals of the team for each sprint at the SPM, or reviewing the goals for the sprint in the SRRM. Obtaining feedback from the customer was a slow process and took place outside of the meetings. Therefore, it was not possible for the customer to adopt the common goals, norms and values of the team, which according to the literature results in a lack of identification with the team (Eckel and Grossman, 2005). It also made it difficult for the team to develop a common identity with the customer, which was evident from team members as they did not consider the customer part of the team.

The SRRM was the weakest of the three agile practices for developing a sense of identity, which may be a consequence of its implementation. C2 was the only case where team members reported a very positive experience of the practice as team members interacted and contributed to the meeting with feedback received from the customer on the software delivered. The team in C3 obtained similar benefits from the practice, but it was recognised that in this team the practice could be improved by increased identification of areas for improvement and associated actions, whereas the benefit from the SRRM in C1 was almost non-existent. It was difficult to see how the SRRM could contribute to a sense of identity within C1 as little occurred at this meeting that would help team members feel part of the team, or work cooperatively together. As this meeting only occurred once every three weeks it is likely that this may only be a small factor where there is a reduced sense of identity.

7.2.2 Members exhibit a strong sense of commitment to the clan

The SPM and the DSM encouraged team members across all three teams to exhibit a strong sense of commitment to the team. In the SPM team members had a high level of involvement in defining and planning the short-term goals for each sprint, which consistent with the literature contributes to a higher level of commitment (Bishop et al., 2000; Buchanan, 1974). The SPM required a great deal of interaction and communication with team members agreeing on the common goal to which they collectively committed to deliver at the end of the sprint, and team members feeling personally responsible for the tasks that they were assigned, which is consistent with the development of commitment within a team (Katzenbach and Smith, 2005). The DSM also required daily interaction and communication with team members aspiring to achieve their personal goals on a daily basis. It also provided visibility and

transparency on daily goals and progress against those goals with team members willing to expend additional effort, such as working additional time to complete their daily tasks, or to prepare for the SPM. This too is consistent with the literature where a sense of commitment is fostered by having a clearly defined goal, involving all team members in all phases of the project, resulting in a willingness of team members to expend additional effort (Larson and LaFasto, 1989, p74).

The commitment of the customer to the team is considered critical for the success of an agile project (Chow and Cao, 2008; Misra et al., 2009; Nerur and Balijepally, 2007). In C1 the customer was not considered part of the team and did not participate in the SPM or the DSM. The conclusion that can be drawn from this is that their lack of participation indicated a lack of commitment to the team and to the project from the perspective of the team, which contrasted dramatically with the customer participation in both C2 and C3, where the commitment of the customer to the team was clearly evident as they participated regularly in the SPM and the DSM and provided input, advice, direction, and feedback to the team.

There was limited evidence that the SRRM contributed to high levels of commitment across the three teams. This was partially due to the nature of the practice as this meeting was a period of review and reflection for the team and was not a forum for setting goals or assigning tasks to team members for completion. It was also a meeting where the customer reviewed the completed software. In C2 and C3 the customer actively participated in this review, which demonstrated their commitment to the team. However, the limited evidence of commitment may also be due to the poor implementation of the practice as detailed in section 7.2.1. Part of the rationale for the SRRM is to identify areas in which the team can improve in subsequent sprints, which can then become future goals for the team. The lack of identification of such areas in C1 and C3, particularly in C1 could suggest a lack of commitment by the team to improve in the future.

7.2.3 Members develop a strong sense of loyalty to the clan

Each of the three agile practices contributed to the development of a strong sense of loyalty in the three teams, but the DSM provided the strongest evidence for this characteristic. The three practices encouraged greater interaction and communication

amongst team members, which contributed to their sense of belonging to the team as indicated by Whitworth and Biddle (2007b), even though in C1 the participation of team members in the meetings was irregular. These meetings also contributed to the identification of team members with the team as discussed in section 7.2.1, which relates to levels of loyalty and is consistent with the literature (Mak and Sockel, 2001).

QA regularly participated in the SPM in both C1 and C3, which helped to develop loyalty between them and the team. The evidence to support this was stronger in C3 than in C1 due to the participation of QA in the study in C3. It is possible that the same positive experience may not be obtained from the QA team in C1 if they had participated in the study. The team members in C1, based in India, were distributed from the core development team and were part of the team for a number of months. Participation in the SPM required these team members to work additional hours in order to attend the meeting once every three weeks. Lack of attendance was not reported, so it could be assumed that they had some level of loyalty to the team to attend the meeting, but it could also have related to a feeling of obligation on their part to attend, which could be part of their culture. Their culture may also have limited their contribution to the meeting as these team members were described as submissive and they may not have felt in a position to make suggestions or recommendations to the remainder of the team. While the development team did consider QA part of the team, it is unknown if QA themselves felt part of the team. These findings suggest that it can be difficult for distributed team members from different cultures to participate fully in a team, which could impact on their level of loyalty to the team.

Team members in C2 and C3 collaborated with the customer in all aspects of SPM, such as establishing goals, estimating tasks, and assigning tasks, which helped to develop their team sprint as it provided them with autonomy over their workload for the sprint. However, collaboration was not conducted to the same extent in C1 where the Project Manager pre-assigned tasks and asked team members to determine estimates for their assigned tasks. Collaboration in the SPM was limited where tasks and estimates were presented by individuals assigned, but there was little discussion in relation to each task or estimate, and these were rarely changed. This allowed the

development of experts within the team, with those most familiar within a particular area considered the best person for the task and the most appropriate person to provide an estimate. It was possible that other team members did not feel qualified enough to comment on the estimates provided as their expertise did not lie in that area, which resulted in the lack of discussion and lack of collaboration on tasks and estimates. Therefore, learning and knowledge sharing was reduced. This could also have impacted on the team spirit within the team, as the views in this team were mixed with team spirit described as ‘pretty good’, ‘ok’ and ‘medium’. However, it was likely that this was also related to the factors outside of the agile practices as redundancies had occurred within the organisation in the previous year, which had generated uncertainty for employees in relation to the stability of their employment. This suggests that lack of collaboration may be a contributory factor to lower levels of loyalty in a team, but that it is not the only factor.

The DSM provided the strongest evidence for the development of loyalty to the team. Team members interacted and communicated on a daily basis. They worked closely together toward a common goal, were dependent on one other, and provided assistance to each other when required. All of this helped to develop a sense of belonging and solidarity with the team (Alvesson and Lindkvist, 1993). The regular interaction also allowed team members to become more comfortable with each other and become familiar with their competencies. The DSM provided visibility and transparency on tasks with team members in a position to quickly identify where there were difficulties. This practice was implemented in a very positive way in each of the three teams with no evidence of criticism observed. Team members were collaborative and cooperative in nature and willing to help one another. This suggests that where a practice is implemented well and the environment is supportive that it can be a strong contributor to the development of loyalty within a team.

The SRRM was the least beneficial of the three practices in relation to the development of loyalty within the team due to its poor implementation, particularly in C1. Similar to the SPM and the DSM it encouraged regular communication and interaction, which took place at the end of the sprint. Where present in the team, the customer, QA and distributed team members participated in the meeting. The demonstration of the completed software to the customer in C2 and C3 helped to

build loyalty between them and the remainder of the team as the customer could determine if the team had delivered on the goal agreed at the SPM. Where a customer queried an aspect of the software delivered, team members defended the team and justified what they had completed, which occurs when individuals are loyal to a team as indicated by Ang and Slaughter (2001). Praise when received from the customer following the demonstration of working software contributed to the development of a good team spirit. This suggests that where the customer regularly participates in the team, loyalty can develop between the customer and the remainder of the team, which is beneficial for all. It avoids a blame culture or conflict between the customer and the team, which appeared to be the case in C1 where the customer was reported to place blame on the team without actively participating in the team.

7.2.4 Members develop common interests and goals which evolve as tasks progress

The SPM and DSM contributed to the development of common interests and goals in all three cases, whereas the evidence from the SRRM was mixed. This is not unexpected as the SPM and the DSM are meetings where sprint goals and daily goals are defined and agreed, which can evolve from day to day and from sprint to sprint. As indicated by Ouchi (1979) the goals are often unknown at the start of a project where clan control exists. This is true of each sprint as the short-term goals for the sprint are unknown until clarity on the sprint goal is obtained at the SPM, which is common in software development as teams gain a greater understanding of the system as the project progresses (Walz et al., 1993). Goals for the team develop and evolve as members become socialised into the clan and become familiar with the norms and values of the clan (Alvesson and Lindkvist, 1993; Harris et al., 2009), which is illustrated in an ASD team by their continued participation and contribution at the SPM and the DSM. Team members also have autonomy over their own goals and how to achieve them, which provides some evidence that clan control exists (Maruping et al., 2009). In addition, the customer regularly participates in the SPM and the DSM in C2 and C3, which ensures that their goals are aligned with the goals of the team, and the customer can provide feedback to the team as indicated by Aspronni (2004).

Similar to the three characteristics already discussed, the SRRM provided little opportunity for the teams to develop common interests or goals. This was partially due to the nature of the practice, which was predominantly a means for the team to review and reflect on the sprint. However, the team was expected to identify areas for improvement, which could become goals for subsequent sprints, but only C2 proactively observed this practice with some actions identified in C3 and none in C1. This could be related to the interest of the teams in C2 and C3 to use and implement the agile practices as expected by the methodology as both of these teams had recently implemented Scrum and were still learning and adjusting to the new way of developing software. It is also possible that after two years on the same project the team in C1 may feel that there is little they can improve and consequently do not feel the need to identify any actions.

7.2.5 Rituals and ceremonies identify and reinforce acceptable behaviours

The three practices were very strong as rituals and with identifying and reinforcing acceptable behaviours. Each agile practice can be viewed as a ritual and a ceremony by holding the meeting at the same time, in the same place (Schwaber and Beedle, 2002, p119). They occurred regularly and had become habitual in each of the three teams due to the number of instances of each meeting that team members have participated in, with team members automatically gathering at the specified time and location to conduct the practice. Each practice was a source of information for members (Kirsch, 1997; Ouchi, 1979), particularly for new and distributed team members as it allowed them to quickly become familiar with the norms, values and behaviours of the team. Each agile practice allowed teams to develop their own acceptable behaviours, although these behaviours may not always be good practice as was demonstrated in C2 at the DSM (section 5.2.5 p.194). The socialisation of individuals into the team through attendance at and contribution to the practices also allowed team members to reinforce the accepted behaviours (Hegtvedt, 2005; Kirsch, 1997).

7.2.6 Members are influenced by shared norms, values and beliefs

There was supporting evidence that shared norms, values and beliefs determined by each of the agile practices influenced the behaviour of team members, which

demonstrated that clan control was in operation. As indicated by the literature, where common values and beliefs exist, team members are cooperative and share common goals and interests (Birnberg and Snodgrass, 1988; Ouchi, 1980; Wilkins and Ouchi, 1983), which was evident in these three teams as discussed in section 7.2.4. Certain norms had developed when using each of the agile practices, which influenced behaviour.

The norm in C2 and C3 was to attend and contribute to each of the agile practices, which demonstrated that clan control was in operation (Kirsch et al., 2010), but in C1 the norm was to attend if available and to contribute when required. This resulted in differences in shared values across the teams. Learning, sharing of information, cooperation and collaboration were more dominant in C2 and C3 than in C1, which also demonstrated the existence of clan control (Kirsch et al., 2010). It was possible that differences in shared values between the three cases related to the long-term use of the practices in C1 and the development of particular roles or experts in this team. It may also be due to the experience of the team with agile as they regularly communicated outside of the meetings and did not feel the need to wait for a meeting to seek an answer to questions. However, the limited contribution of distributed team members in C1 resulted in additional monitoring by the Project Manager at the SPM and the DSM to ensure that distributed team members were progressing with their tasks as expected. It also resulted in limited learning at the SRRM.

The regular participation of the customer in C2 and C3 allowed them to influence the remainder of the team on their priorities and needs, and ensure that they develop common interests and goals as indicated by Hoda et al. (2011). However, in C3 the presence of management and the customer at the review part of the SRRM influenced the contribution of team members as they felt they could not reflect honestly when management were present. In C1 the lack of customer participation influenced the behaviour of the team in the SPM and the DSM, where the team had started to make assumptions on behalf of the customer as the response time from the customer was very poor. The team had to make assumptions in order to progress with their work, even when they were aware that this could require future rework.

As indicated by Schwaber and Beedle (2002, p119) shared values and beliefs are reinforced through the agile practices, which was evident in each of the three cases. The desire to increase learning in C2 influenced the behaviour of team members to assign each task to two people and to rotate tasks in the SPM. Task rotation also occurred in C3, but in C1 learning was limited as team members were assigned specific tasks relating to their expertise. This likely emerged over time due to pressure to deliver substantial functionality in a tight time period. The use of the Product Backlog and the Scrum Board encouraged team members to be honest in their estimates and their progress as this information was visible to all. However, it was difficult to determine at the SRRM if team members were reflecting honestly on their performance in the sprint. The regular communication and interaction amongst team members also reinforced collaboration and cooperation, which were valued by team members. It is recognised that communicating face-to-face is appropriate for discussing, dealing with, and resolving issues related to software projects (Robey et al., 1989). This requires team members to engage with the process and contribute to the discussion which occurred intermittently in C1 and regularly in C2 and C3. Such participation can also lead to individuals influencing each other in relation to decisions made, or behaviours that may be considered acceptable, and it is suggested that influence is an important factor in the management of ISD projects (Robey et al., 1989).

These findings suggest that the agile practices can be a strong influence on team members and how they operate as a team. In these three teams the influences were mostly positive and they benefited the team. However, it is possible to see how they could be used in a negative way and influence on how team members behave. Each of these teams were willing to embrace agile and appeared to be very cooperative and have good relationships with each other, which possibly impacted on how individuals positively influenced each other.

7.2.7 Members evaluate and reward or sanction each other based on conformance to the values and beliefs of the clan

The evidence to support this characteristic was limited across the three teams, which was unsurprising as evaluation and rewards in all three organisations were conducted through a formal rewards system. C1 was the weakest in terms of this characteristic.

The lack of evaluation, provision of rewards or sanctioning in C1 could be due to the length of time that team members were using the practice and the long duration of the project where team members do not feel the need to reward or sanction each other in any way as individuals are aware of what was expected of them. However, there was some evidence of evaluation and sanctioning at the SPM in C1, but this only related to minor non-conformance of individuals to the norms of the team.

The DSM provided the greatest support for this characteristic, but evidence was not particularly strong with team members evaluating each other against their personal daily goals or evaluating each other against conformance or non-conformance to the norms of the team. There was evidence of rewards in C3, and only some evidence of rewards or sanctioning in C2. Where team members did not achieve their daily goals there was little appetite in any of the teams to criticise or sanction individuals. Instead, help was provided to complete tasks, if required. Other informal sanctions such as sarcasm, or a disapproving look (Kirsch, 2004; Westphal and Khanna, 2003) were not apparent during data collection, but it was possible that these were used on occasion, but were not obvious during the limited amount of time the researcher was present for data collection. Rewards in C2 and C3 were informal, such as verbal praise from team members, or from the customer, which is consistent with the literature (Choudhury and Sabherwal 2003; Kirsch et al., 2002; Kirsch, 2004), but was not evident in C1. It is possible that the verbal rewards evident in C2 and C3 related to the length of time since the implementation of Scrum. As team members were relatively new to the practices in comparison to C1, it is likely that team members felt the need to acknowledge effort.

As little evidence was determined in relation to this characteristic, this suggests that the subtle nature of evaluation in clans (Ouchi, 1980) made it difficult to identify if teams were evaluating and rewarding or sanctioning each other to any great extent. Where evaluation was apparent it was identified, but other occurrences may have gone unnoticed because the researcher was not part of the clan or did not observe the teams for a longer period of time, which is recommended in future research (section 7.5).

7.3 Contributions to Research and Practice

This section summarises the contributions and implications of this study to research on clan control and to research in ASD teams. It further highlights implications for practice.

7.3.1 Contributions to Research

This research contributes to the literature in several ways:-

- Clan control is often studied in a broader context in combination with other forms of control and is rarely studied in isolation, which limits the understanding of clan control and its influence in various settings, such as project teams. This study attempts to address this by examining clan control in one particular setting, that of an ASD team.
- This study adds to the control literature by focusing specifically on clan control and presenting a framework for clan control on the pre-conditions required to develop and implement clan control and the resultant characteristics of clan control. This framework was informed by the literature, but is not limited to ASD and provides a foundation for future empirical research on clan control.
- This study builds on prior research of control in ISD (Henderson and Lee, 1992; Kirsch, 1997; Kirsch and Cummings, 1996; Kirsch et al., 2010; Rustagi et al., 2008) and control in ASD (Harris et al., 2009; Maruping et al., 2009) by applying control theory to ASD teams. While the application of control theory in ISD is growing, there is limited application of control theory in an agile context. It adds to the substantial body of literature on agile methodologies and ASD with the first known study of clan control in ASD teams.
- Much empirical research in the area of ASD focuses largely on agile methodologies, rather than on specific agile practices. The study further provides a contribution towards a richer understanding of agile practices and how they can facilitate clan control in ASD teams; for example, by helping to build loyalty and commitment.

7.3.2 Contribution to Practice

This research contributed to practice as follows:

- It provides an insight for practitioners on how ASD teams implement agile practices in different contexts and how the implementation is unique to each team setting, but it also illustrates how other factors can affect the agile practices, such as the distribution of teams or the length of time that the practice is in use.
- It can also provide an insight for managers on how clan control can develop within ASD teams and the impact this may have on the team. This may influence their decision to implement a particular agile practice, or to adapt their current implementation of the practice to encourage clan control within the team. It may also impact on their recruitment and selection process by ensuring that individuals employed have similar values and beliefs to the team and behave in a manner that conforms to the norms of the team. It also highlights that where formal controls exist and are used extensively in teams that it is possible that these can be reduced.
- It highlights that teams should implement the core elements of the practices as recommended by the methodology in order to obtain their full benefit, such as including the customer in the team and ensuring that all team members participate. But, it also suggests that the value of the practices may diminish over time as teams become complacent and they may overlook the original purpose of the practice.
- It also identifies key issues and conditions under which the agile practices contribute to clan control and how they can positively or negatively impact a team e.g. increased communication and interaction amongst individuals, development of unusual norms or behaviours, or a lack of customer participation. This should be of benefit for management to develop an awareness of such issues or conditions so they can create an environment that supports clan control.
- The study concludes with recommendations on how the agile practices can be used to achieve clan control that may help ASD teams to identify shortcomings

in how they have implemented a particular agile practice so that such teams can put mechanisms in place to overcome these shortcomings.

7.4 Limitations

This study has a number of shortcomings, which are highlighted here. They first relate to the limitations of the study itself and secondly to limitations of the research approach.

7.4.1 Limitations of the Study

The following limitations to the study were identified:

- This study was limited to an investigation of two agile methodologies, XP and Scrum. This was intentional to bound the research and also to allow for a more in-depth focus on specific agile practices, which are not necessarily practices of other agile methodologies.
- While many agile practices exist, three practices were selected for investigation in this study. This was deliberate in order to bound the study. It is possible that other agile practices may also help to achieve clan control in different ways, which is an area for future research. Other research should also consider how agile methodologies in general help to achieve clan control, without limiting the study to specific agile practices.
- The study focused solely on the views of the team and did not consider the views of management, who are external to the team. Their views may provide a different perspective on how the agile practices helped to achieve clan control in ASD teams.
- The study was limited by the number of individuals interviewed in each role. For example, two Scrum Masters, two Product Owners, and one QA. While all personnel were interviewed the experiences of each of these roles are not necessarily reflective of the experiences of personnel in other teams. In addition, while one team was distributed, only one distributed team member was interviewed, so his view while valid is only representative of his experience and cannot be considered representative of other distributed team members. Future

studies should incorporate views of multiple distributed team members to see how this parameter would influence results.

- Many researchers have found that different control modes affect each other and are combined in a number of different ways. Examining a single control in isolation may not be truly representative of the control mode and may distort the true extent to which clan control resulted in the teams studied. Therefore, a limitation of this research relates to its focus on a single control mode.
- A number of characteristics detailed in the framework studied were similar and it was difficult to clearly distinguish between each characteristic as the literature provided a number of similar, but slightly different definitions for each characteristic. This caused difficulty when analysing the data as there was a certain degree of overlap between the different characteristics, which is acknowledged as a limitation of the research.
- The framework used in this study contained many different concepts, each of which could be examined in greater detail in a single piece of research. Due to the number of concepts and time restrictions for each interview it was not possible to examine each concept in great detail, which could be an avenue of further research with each concept a separate study on its own.

7.4.2 Limitations of the Research Approach

The following limitations to the research approach were identified:

- One of the main concerns with case study research is the generalisability of the research findings as discussed in section 4.4.5 (p.96). As this study is exploratory it is not attempting to generalise the findings, but rather to present the uniqueness of each case and identify where there are similarities and differences across the teams studied. This study used a multiple-case study approach by examining three teams each in their own natural setting. While the small number of cases studies may be considered a limitation in the context of an exploratory study this number was deemed sufficient as a point of ‘theoretical saturation’ had been reached.

- For the purposes of this research the boundary of the study was the period of time during which data was collected, which can be considered a limitation. As data was collected at a single point in time, this sets a frame of reference for the study and also reflects the perspective of participants at that point in time (Pettigrew, 1990). Team composition may change over time, individuals gain more knowledge and experience and organisations may evolve and change, observing the teams and interviewing team members over a longer period of time may have provided additional insights that were not evident at the time of data collection.
- Data collection in C1 was conducted at intervals over a period of four months, whereas data collection in C2 and C3 occurred over a number of consecutive days. This allowed the researcher to become familiar, to a certain extent, with the behaviours of the teams in C2 and C3 on a day-to-day basis; for example, starting times, lunch breaks and coffee breaks, their work environment and level of interaction with each other. In hindsight this research would have benefited from conducting a similar exercise in C1 to become more familiar with their work environment, and to observe the interactions and behaviours of individuals in their natural setting.
- As with any data collection method, the use of interviews has its limitations as discussed in section 4.5.1 (p.101), which may have introduced bias on the part of the researcher. Also, as the concept under investigation was sensitive, it is possible that participants were not forthcoming with specific information. Details on how this was dealt with are provided in 4.7.1 (p.119). In addition, as many of the concepts studied were subtle or intangible it is possible that they were not apparent to the researcher when observing the team.
- The use of observation as a data collection method has its limitations as discussed in section 4.5.2 (p.104). The number of observations was limited in each of the cases studied and the study may have benefited from additional observations. It is possible that something observed was not noted or not considered important during the observation because it was only observed once, or that an observation was missed altogether. Additional observations could

provide greater insight into how the teams implemented the agile practices. Also, a different researcher may have noted other observations, which they considered important.

- The presence of the researcher may have affected the behaviour of the team when observing the agile practices. If the researcher was present for a longer period of time and developed stronger relationships with team members this impact could be negated.
- A final limitation relates to potential cultural issues due to the location of the team or the culture of the participants in each team. Fourteen of the 25 participants in the study were non-native English speakers. It is possible that the different culture in C2 (team based in Sweden) affected the way in which the agile practices were implemented. It is also possible that some individuals may have misinterpreted a concept, or were not able to communicate a response as effectively as they would if speaking in their native language. Details on how this was addressed are provided in section 4.7.2 (p.123). There was also a limitation conducting a conference call with the distributed team member as it was more difficult to build a rapport with this person and the data collected was not as rich as those in the face-to-face interviews.

Despite these limitations the results of the study provide some interesting insights on how agile practices contribute to clan control in ASD teams.

7.5 Recommendations for Future Research

The results of this study suggest a number of avenues for future research, which are detailed as follows:

- Additional case studies should be conducted to determine if the findings from this study are replicated across other cases in different industries, in teams of different sizes with varying levels of team experience, or varying lengths of time since the implementation of their selected agile methodology.
- A number of ASD teams within the same organisation should be studied in detail to establish if the same findings exist within the context and culture of the same

organisation, or whether they are specific to particular teams within the organisation.

- Additional research on how agile methodologies contributed to the development of clan control in ASD teams would be beneficial to obtain a greater insight on how clan control develops within such teams as a result of the various agile practices that are implemented by the teams
- Future studies could examine the agile practices from the perspective of managers. As they are external to the ASD team, but are closer to the team than an independent researcher, they may have different insights on how the agile practices help to achieve clan control. Also, they may observe actions such as behaviours or norms that emerge within the teams as a consequence of using the agile practices, which were not present prior to their implementation.
- Other studies may compare and contrast findings of co-located and distributed teams. While this study did contain a distributed team, only one distributed team member participated in the study, so it was not possible to do a reasonable comparison across the teams to determine similarities and differences between these two types of teams. Future studies should examine teams consisting of several distributed members.
- As clan control is a subtle and intangible form of control a longitudinal study would allow a more thorough examination of the agile practices and how they facilitated the development of clan control over a period of time. This would allow a researcher to observe the behaviours that emerge within teams as they become more familiar with the practices, or where the practices are adapted to suit the changing needs of the team.
- A survey of all team members across a wide range of teams within a single organisation would be beneficial for future research to determine if generalisations can be inferred across all teams within one organisation and provide an explanation for the emergence of clan control in ASD teams. Similarly, a survey across a number of organisations could verify if findings were generalisable to all agile teams.

- Finally, as the conceptual framework contained a large number of concepts (Figure 3.5-1 on p.77) there was a trade-off between investigating a smaller number of concepts or a smaller number of practices. A decision was made to focus on the concepts rather than the number of agile practices. Therefore, future studies may investigate how other agile practices facilitate clan control, or how other agile practices facilitate other control modes within teams.

7.6 Conclusion

The purpose of this study was to enhance the understanding of clan control in ASD teams. Using control theory as a foundation, this study presented a conceptual framework for clan control comprising the conditions that encourage the development of clans, the conditions that must be in place to implement clan control, both of which help to achieve the resultant characteristics of clan control. Each of these resultant characteristics reinforces or strengthens the development of a clan. This framework was examined in the context of ASD teams in an attempt to gain an understanding of how three agile practices: the sprint planning meeting, the daily scrum meeting, and the sprint review and retrospective meeting can help achieve clan control in ASD teams. In addition, for teams who lack knowledge in the area of clan control, it makes recommendations on how agile practices can improve the development of clan control in ASD teams; for example, by ensuring that their behaviours are consistent with the shared goals, norms and values of the team, reducing requirements for formal controls, and encouraging a sense of collegiality between team members, which will ideally result in a more successful team. It concludes that each of the three agile practices studied helped to achieve clan control in ASD teams, but they are not the sole contributor, as there are other factors, such as the culture of the organisation, the experience of the team, the location of the team, the manner in which the agile practices are implemented, which can also help to achieve clan control. It is also possible that these agile practices could detract from clan control, although there was little evidence in the three cases studied that this occurred. The three teams studied were very cooperative in nature, but if teams studied contained members who were very competitive, the findings may be quite different. These findings provide much needed empirical evidence about clan control in ASD teams, which is important for the development of research on

informal control modes and particularly in ASD where the emphasis on traditional, bureaucratic controls is reduced.

References

- Abbas, N., Gravell, A. M. and Wills, G. B. (2010), Using Factor Analysis to Generate Clusters of Agile Practices (A Guide for Agile Process Improvement), AGILE Conference, 2010, Orlando, Florida, 9-13 August, pp. 11-20
- Abernethy, M. A. and Stoelwinder, J. U. (1995) The role of professional control in the management of complex organizations, Accounting, Organizations and Society, Vol. **20**(1), pp. 1-17.
- Abrahamsson, P., Conboy, K. and Wang, X. (2009) Lots done, more to do: The current state of agile systems development research, European Journal of Information Systems, Vol. **18**(4), pp. 281-284.
- Abrahamsson, P., Salo, O., Ronkainen, J. A. and Warsta, J. (2002) *Agile Software Development: Review and Analysis*, VTT Publications 478.
- Abrahamsson, P., Warsta, J., Siponen, M. T. and Ronkainen, J. A. (2003), New Directions on Agile Methods: A Comparative Analysis, Software Engineering, 2003. Proceedings. 25th International Conference on, Portland, Oregon, USA, pp. 244-254
- Adler, P. S. (2001) Market, Hierarchy, and Trust: The Knowledge Economy and the Future of Capitalism, Organization Science, Vol. **12**(2), pp. 215-234.
- Ågerfalk, P. J. and Fitzgerald, B. (2006) Flexible and distributed software processes: Old petunias in new bowls?, Communications of the ACM, Vol. **49**(10), pp. 27-34.
- AgileAlliance (2001) "Manifesto for Agile Software Development" [Online] [Accessed 12th December, 2007] Available from the World Wide Web: www.agilemanifesto.org,
- Alvesson, M. and Lindkvist, L. (1993) Transactions Costs, Clans and Corporate Culture, Journal of Management Studies, Vol. **30**(3), pp. 427-452.
- Ang, S. and Slaughter, S. A. (2001) Work Outcomes and Job Design for Contract Versus Permanent Information Systems Professionals on Software Development Teams, MIS Quarterly, Vol. **25**(3), pp. 321-350.
- Angioni, M., Carboni, D., Pinna, S., Sanna, R., Serra, N. and Soro, A. (2006) Integrating XP Project Management in Development Environments, Journal of Systems Architecture, Vol. **52**(11), pp. 619-626.
- Asproni, G. (2004) Motivation, Teamwork, and Agile Development, Agile Times, Vol. **4**(1), pp.
- Atlas, A. (2009), Accidental Adoption: The Story of Scrum at Amazon.com, Agile Conference, 2009. AGILE '09., Chicago, IL, 24-28 August, 2009, pp. 135-140
- Augustine, S., Payne, J. H., Sencindiver, F. and Woodcock, S. (2005) Agile Project Management: Steering from the Edges, Communications of the ACM, Vol. **48**(12), pp. 85-89.
- Barker, J. R. (1993) Tightening the Iron Cage: Concertive Control in Self-Managing Teams, Administrative Science Quarterly, Vol. **38**(3), pp. 408-437.
- Barnett, L. and Schwaber, C. (2004) Adopting Agile Development Processes Improve Time-To-Benefits For Software Projects, Forrester Report, Vol.
- Baumard, P. (1999) *Tacit knowledge in organizations*, SAGE Publications, Thousand Oaks, California: .
- Beck, K. (2000) *Extreme Programming Explained: Embrace Change*, Addison-Wesley, Boston, MA.
- Beck, K. and Andres, C. (2005) *Extreme Programming Explained: Embrace Change*, 2nd ed., Addison-Wesley, Boston, MA.
- Beck, K. and Boehm, B. (2003) Agility through Discipline: A Debate, Computer, Vol. **36**(6), pp. 44-46.
- Becker, T. E. and Billings, R. S. (1993) Profiles of commitment: An empirical test, Journal of Organizational Behavior, Vol. **14**(2), pp. 177-190.

- Benbasat, I., Goldstein, D. and Mead, M. (1987) The Case Research Strategy in Studies of Information Systems., MIS Quarterly, Vol. **11**(3), pp. 369-386.
- Berry, A. J., Broadbent, J. and Otley, D. T. (2005) *Management Control: Theories, Issues and Performance*, 2nd edn., Palgrave Macmillan, Basingstoke, England.
- Berry, A. J., Coad, A. F., Harris, E. P., Otley, D. T. and Stringer, C. (2009) Emerging themes in management control: A review of recent literature, *The British Accounting Review*, Vol. **41**(1), pp. 2-20.
- Birnberg, J. G. and Snodgrass, C. (1988) Culture and control: A field study, *Accounting, Organizations and Society*, Vol. **13**(5), pp. 447-464.
- Bishop, J. W., Scott, K. D. and Burroughs, S. M. (2000) Support, commitment, and employee outcomes in a team environment, *Journal of Management*, Vol. **26**(6), pp. 1113-1132.
- Blau, P. M. and Scott, W. R. (1963) *Formal Organizations: A Comparative Approach*, Routledge and Kegan Paul Ltd., London, UK.
- Boehm, B. (1988) A spiral of software development and enhancement, *IEEE Computer*, Vol. **21**(5), pp. 61-72.
- Boehm, B. (2002) Get Ready for Agile Methods, With Care, *IEEE Computer*, Vol. **35**(1), pp. 64-69.
- Boehm, B. and Turner, R. (2005) Management Challenges to Implementing Agile Processes in Traditional Development Organizations, *IEEE Software*, Vol. **22**(5), pp. 30-39.
- Bose, I. (2008) Lessons Learned from Distributed Agile Software Projects: A Case-Based Analysis, *Communications of the AIS*, Vol. **23**(15), pp. 619-632.
- Braithwaite, K. and Joyce, T. (2005) XP Expanded: Distributed Extreme Programming In *Extreme Programming and Agile Processes in Software Engineering (XP 2005)*, 3556 Springer Berlin / Heidelberg, pp. 1524-1526.
- Buchanan, B. (1974) Building Organizational Commitment: The Socialization of Managers in Work Organizations, *Administrative Science Quarterly*, Vol. **19**(4), pp. 533-546.
- Cardinal, L. B. (2001) Technological Innovation in the Pharmaceutical Industry: The Use of Organizational Control in Managing Research and Development, *Organization Science*, Vol. **12**(1), pp. 19-36.
- Carmeli, A., Gelbard, R. and Goldreich, R. (2005) Linking perceived external prestige and collective identification to collaborative behaviors in R&D teams, *Expert systems with applications*, Vol. **38**(7), pp. 8199-8207.
- Carver, C. S. and Scheier, M. F. (1999) Control Theory: A Useful Conceptual Framework for Personality-Social, Clinical and Health Psychology In *The Self in Social Psychology*, Psychology Press, pp. 299-316.
- Cavaye, A. L. M. (1996) Case Study Research: A Multi-Faceted Research Approach for IS, *Information Systems Journal*, Vol. **6**(3), pp. 227-242.
- Ceschi, M., Sillitti, A., Succi, G. and De Panfilis, S. (2005) Project Management in Plan-Based and Agile Companies, *Software, IEEE*, Vol. **22**(3), pp. 21-27.
- Chen, W. and Hirschheim, R. (2004) A paradigmatic and methodological examination of information systems research from 1991 to 2001., *Information Systems Journal*, Vol. **14**(3), pp. 197-235.
- Child, J. and McGrath, R. G. (2001) Organizations Unfettered: Organizational Form in an Information-Intensive Economy, *Academy of Management Journal*, Vol. **44**(6), pp. 1135-1148.
- Choudhury, V. and Sabherwal , R. (2003) Portfolios of Control in Outsourced Software Development Projects, *Information Systems Research*, Vol. **14**(3), pp. 291-314.
- Chow, T. and Cao, D.-B. (2008) A Survey Study of Critical Success Factors in Agile Software Projects, *Journal of Systems and Software*, Vol. **81**(6), pp. 961-971.
- Chriss, J. (2007) *Social Control: An Introduction*, Polity Press, Cambridge, UK.
- Chua, C., Lim, W. K., Soh, C. and Sia, S.-K. (2007), Building Swift Clan Control in Complex IT Projects, International Research Workshop on IT Project Management 2007, Montréal, Québec, Canada, December 8th 2007, pp. Paper 1

- Cockburn, A. (2004) *Crystal Clear: A Human-Powered Methodology for Small Teams* Addison Wesley, Boston, MA.
- Cockburn, A. (2007) *Agile Software Development: The Cooperative Game*, 2nd edn., Addison Wesley, NJ, USA.
- Cockburn, A. and Highsmith, J. (2001) Agile Software Development: The People Factor, IEEE Computer, Vol. **34**(11), pp. 131-133.
- Cohen, S. G. and Bailey, D. E. (1997) What makes teams work: Group effectiveness research from the shop floor to the executive suite, Journal of Management, Vol. **23**(3), pp. 239-290.
- Cohn, M. and Ford, D. (2003) Introducing an Agile Process to an Organization, IEEE Computer, Vol. **36**(6), pp. 74-78.
- Collier, P. M. (2005) Entrepreneurial control and the construction of a relevant accounting, Management Accounting Research, Vol. **16**(3), pp. 321-339.
- Conboy, K. (2009) Agility from first principles: Reconstructing the concept of agility in information systems development, Information Systems Research, Vol. **20**(3), pp. 329-354.
- Conboy, K. and Fitzgerald, B. (2010) Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion, ACM Transactions on Software Engineering and Methodology Vol. **20**(1), pp. Article 2.
- Connell, J. and Brice, L. (1984) Rapid Prototyping Datamation, Vol. **30**93-100.
- Cooper, D. R. and Schindler, P. S. (2003) *Business Research Methods*, 8th edition, McGraw-Hill/Irwin, Boston ; London
- Coram, M. and Bohner, S. (2005), The Impact of Agile Methods on Software Project Management, Engineering of Computer-Based Systems, 2005. ECBS '05. 12th IEEE International Conference and Workshops on the, pp. 363-370
- Corbin, J. and Strauss, A. (2008) *Basics of Qualitative Research*, 3rd edition, Sage Publications, London, UK.
- Covaleski, M. A., Dirsmith, M. W., Heian, J. B. and Samuel, S. (1998) The Calculated and the Avowed: Techniques of Discipline and Struggles Over Identity in Big Six Public Accounting Firms, Administrative Science Quarterly, Vol. **43**(2), pp. 293-327.
- Cram, W. A. and Brohman, M. K. (2010), Beyond Modes: A New Typology of ISD Control, International Conference on Information Systems (ICIS 2010), December 12-15th, St. Louis, MO, pp. Paper 94
- Creswell, J. W. (2003) *Research Design: Qualitative, Quantitative and Mixed Methods Approaches*, 2nd edition, Sage Publications, Thousand Oaks, CA.
- Daft, R. L. (2009) *Organization Theory and Design* 10th edn, South-Western, Division of Thomson Learning.
- Dahlberg, H., Ruiz, F. S. and Olsson, C. M. (2006) In *IFIP International Federation for Information Processing*, Vol. 206 (Eds, Donnellan, B., Larsen, T. J., Levine, L. and DeGross, J. I.), pp. 291-312.
- Dalton, D. R., Daily, C. M. and Wimbush, J. C. (1997) Collecting "Sensitive" Data in Business Ethics Research: A Case for the Unmatched Count Technique (UCT), Journal of Business Ethics, Vol. **16**(10), pp. 1049-1057.
- Das, T. K. and Teng, B.-S. (1998) Between Trust and Control: Developing Confidence in Partner Cooperation in Alliances, The Academy of Management Review, Vol. **23**(3), pp. 491-512.
- Das, T. K. and Teng, B.-S. (2001) Trust, Control, and Risk in Strategic Alliances: An Integrated Framework, Organization Studies, Vol. **22**(2), pp. 251-283.
- DeMarco, T. (1982) *Controlling Software Projects: Management Measurement and Estimation.*, Prentice-Hall, Englewood Cliffs, NJ.
- Dransfield, P. (1968) *Engineering systems and automatic control*, Englewood Cliffs, N.J., Prentice-Hall.
- Drucker, P. F. (1988) The coming of the new organization, Harvard Business Review, Vol. **66**(1), pp. 45-53.

- Druskat, V. U. and Wheeler, J. V. (2003) Managing from the Boundary: The Effective Leadership of Self-Managing Work Teams, *Academy of Management Journal*, Vol. **46**(4), pp. 435-457.
- Dubé, L. and Paré, G. (2003) Rigor in Information Systems positivist case research: Current practices, trends, and recommendations., *MIS Quarterly*, Vol. **27**(4), pp. 597-635.
- Dybå, T. and Dingsøyr, T. (2008) Empirical Studies of Agile Software Development: A Systematic Review, *Information and Software Technology*, Vol. **50**(9-10), pp. 833-859.
- Eckel, C. C. and Grossman, P. J. (2005) Managing diversity by creating team identity, *Journal of Economic Behavior & Organization*, Vol. **58**(3), pp. 371-392.
- Eisenhardt, K. M. (1985) Control: Organizational and Economic Approaches, *Management Science*, Vol. **31**(2), pp. 134-149.
- Eisenhardt, K. M. (1989) Building Theories from Case Study Research, *The Academy of Management Review*, Vol. **14**(4), pp. 532-550.
- Elssamadisy, A. (2007) *Patterns of Agile Practice Adoption: The Technical Cluster*, C4Media, USA.
- Elssamadisy, A. (2008) *Agile Adoption Patterns: A Roadmap to Organizational Success*, Addison Wesley, USA.
- Emmanuel, C., Otley, D. and Merchant, K. (1990) *Accounting for Management Control*, 2nd edn., Chapman & Hall, London.
- Engelberg, S. (2005) *A mathematical introduction to control theory* London : Imperial College Press ; Singapore ; Hackensack, NJ.
- ExtremeProgramming (2009a) Daily Stand-Up Meeting [Online], [Accessed 11th January, 2011] Available from the World Wide Web:
<http://www.extremeprogramming.org/rules/standupmeeting.html>
- ExtremeProgramming (2009b) How do I start this XP thing? [Online], [Accessed 11th January, 2011] Available from the World Wide Web:
<http://www.extremeprogramming.org/start.html>
- ExtremeProgramming (2009c) Iteration Planning Meeting [Online], [Accessed 12th January, 2011] Available from the World Wide Web:
<http://www.extremeprogramming.org/rules/iterationplanning.html>
- Faraj, S. and Sproull, L. (2000) Coordinating Expertise in Software Development Teams, *Management Science*, Vol. **46**(12), pp. 1554-1568.
- Feldman, M. S. and March, J. G. (1981) Information in Organizations as Signal and Symbol, *Administrative Science Quarterly*, Vol. **26**(2), pp. 171-186.
- Ferreira, A. and Otley, D. (2005) The design and use of performance management systems: An extended framework for analysis, *Management Accounting Research*, Vol. **20**(4), pp. 263-282.
- Fitzgerald, B., Hartnett, G. and Conboy, K. (2006) Customising Agile Methods to Software Practices at Intel Shannon, *European Journal of Information Systems*, Vol. **15**(2), pp. 197-210.
- Fitzgerald, B., Russo, N. and O'Kane, T. (2003) Software Development Method Tailoring at Motorola, *Communications of the ACM*, Vol. **46**(4), pp. 64-70.
- Fitzgerald, B., Russo, N. and Stolterman, E. (2002) *Information Systems Development : Methods-in-Action* McGraw-Hill Higher Education.
- Flamholtz, E. G., Das, T. K. and Tsui, A. S. (1985) Toward an Integrative Framework of Organizational Control, *Accounting, Organizations and Society*, Vol. **10**(1), pp. 35-50.
- Fortado, B. (1994) Informal Supervisory Social Control Strategies, *Journal of Management Studies*, Vol. **31**(2), pp. 251-274.
- Galliers, R. (1992) *Information Systems Research: Issues, Methods and Practical Guidelines*, Blackwell Scientific Publications, Oxford, England.
- Giglioni, G. B. and Bedeian, A. G. (1974) A Conspectus of Management Control Theory 1900-1972, *Academy of Management Journal*, Vol. **17**(2), pp. 292-305.

- Gopal, A. and Gosain, S. (2010) The Role of Organizational Controls and Boundary Spanning in Software Development Outsourcing: Implications for Project Performance Information Systems Research, Vol. **21**(4), pp. 960-982.
- Govindarajan, V. and Fisher, J. (1990) Strategy, Control Systems, and Resource Sharing: Effects on Business-Unit Performance, Academy of Management Journal, Vol. **33**(2), pp. 259-285.
- Gremillion, L. L. and Pyburn, P. (1983) Breaking the systems development bottleneck, Harvard Business Review, Vol. **61**(2), pp. 130-137.
- Guba, E. G. and Lincoln, Y. S. (1994) Competing Paradigms in Qualitative Research In *Handbook of Qualitative Research*, (Eds, Denzin, N. K. and Lincoln, Y.) Thousand Oaks, CA, pp. 105-117.
- Guba, E. G. and Lincoln, Y. S. (1998) Competing Paradigms in Qualitative Research In *The Landscape of Qualitative Research*, (Eds, Denzin, N. K. and Lincoln, Y.) Thousand Oaks, CA, pp. 195-220.
- Guzzo, R. A. and Dickson, M. W. (1996) Teams in Organisations: Recent research on Performance and Effectiveness, Annual Review of Psychology, Vol. **47** 307-338.
- Hackman, J. R. (1990) *Groups that Work (and those that don't). Creating Conditions for Effective Teamwork*, Jossey-Bass Inc. San Francisco, CA.
- Hansson, C., Dittrich, Y., Gustafsson, B. and Zarnak, S. (2006) How Agile are Industrial Software Development Practices?, Journal of Systems and Software, Vol. **79**(9), pp. 1295-1311.
- Harris, M. L., Collins, R. W. and Hevner, A. R. (2009) Control of Flexible Software Development Under Uncertainty, Information Systems Research, Vol. **20**(3), pp. 400-419.
- Hegtvedt, K. A. (2005) Doing Justice to the Group: Examining the Roles of the Group in Justice Research, Annual Review of Sociology, Vol. **31**(1), pp. 25-45.
- Henderson, J. C. and Lee, S. (1992) Managing I/S Design Teams: A Control Theories Perspective, Management Science, Vol. **38**(6), pp. 757-777.
- Herath, S. K. (2007) A framework for management control research, Journal of Management Development, Vol. **26**(9), pp. 895-915.
- Herzberg, F. (1968) One more time: How do you motivate employees?, Harvard Business Review, Vol. **46**(1), pp. 53-62.
- Highsmith, J. and Cockburn, A. (2001) Agile Software Development: The Business of Innovation, IEEE Computer, Vol. **34**(9), pp. 120-122.
- Hirschheim, R. (1992) Information Systems Epistemology: An Historical Perspective In *Information Systems Research: Issues, Methods and Practical Guidelines* Blackwell Scientific, Oxford, England, pp. 28-60.
- Hoda, R., Noble, J. and Marshall, S. (2011) The impact of inadequate customer collaboration on self-organizing agile teams, Information and Software Technology, Vol. **53**(5), pp. 521-534.
- Hopwood, A. G. (1974) *Accounting and Human Behaviour*, Prentice-Hall, London, UK.
- Ilgen, D., Fisher, C. and Taylor, M. (1979) Consequences of individual feedback on behavior in organizations, Journal of Applied Psychology, Vol. **64**(4), pp. 349-371.
- Jacobs, O. L. R. (1974) *Introduction to Control Theory*, Oxford : Clarendon Press.
- Jaworski, B. J. (1988) Toward a Theory of Marketing Control: Environmental Context, Control Types, and Consequences, Journal of Marketing, Vol. **52**(3), pp. 23-39.
- Jaworski, B. J., Stathakopoulos, V. and Krishnan, H. S. (1993) Control Combinations in Marketing: Conceptual Framework and Empirical Evidence Journal of Marketing, Vol. **57**(1), pp. 57-69.
- Jick, T. D. (1979) Mixing Qualitative and Quantitative Methods: Triangulation in Action, Administrative Science Quarterly, Vol. **24**(4), pp. 602-611.
- Jurison, J. (1999) Software Project Management: The Manager's View, Communications of the AIS, Vol. **2**(17), pp. 1-50.

- Kanter, R. M. (1968) Commitment and Social Organization: A Study of Commitment Mechanisms in Utopian Communities, *American Sociological Review*, Vol. **33**(4), pp. 499-517.
- Kaplan, B. and Duchon, D. (1988) Combining Qualitative and Quantitative Methods Information Systems Research: A Case Study, *MIS Quarterly*, Vol. **12**(4), pp. 571-586.
- Karlsson, F. and Agerfalk, P. (2009) Exploring agile values in method configuration, *European Journal of Information Systems*, Vol. **18**(4), pp. 300-316.
- Katzenbach, J. R. and Smith, D. K. (2005) The Discipline of Teams, *Harvard Business Review*, Vol. **83**(7/8), pp. 162-171.
- Kirsch, L. J. (1996) The Management of Complex Tasks in Organizations: Controlling the Systems Development Process, *Organization Science*, Vol. **7**(1), pp. 1-21.
- Kirsch, L. J. (1997) Portfolios of Control Modes and IS Project Management, *Information Systems Research*, Vol. **8**(3), pp. 215-239.
- Kirsch, L. J. (2004) Deploying Common Systems Globally: The Dynamics of Control, *Information Systems Research*, Vol. **15**(4), pp. 374-395.
- Kirsch, L. J. and Cummings, L. L. (1996) Contextual Influences on Self-Control of IS Professionals Engaged in Systems Development, Accounting, Management and Information Technologies, Vol. **6**(3), pp. 191-219.
- Kirsch, L. J., Ko, D.-G. and Haney, M. H. (2010) Investigating the Antecedents of Team-Based Clan Control: Adding Social Capital as a Predictor, *Organization Science*, Vol. **21**(2), pp. 469-489.
- Kirsch, L. J., Sambamurthy, V., Ko, D.-G. and Purvis, R. L. (2002) Controlling Information Systems Development Projects: The View from the Client, *Management Science*, Vol. **48**(4), pp. 484-498.
- Klein, G. and Jiang, J. J. (2001) Seeking consonance in information systems, *Journal of Systems and Software*, Vol. **56**(2), pp. 195-202.
- Klein, H. K. and Myers, M. D. (1999) A set of principles for conducting and evaluating interpretive field studies in information systems, *MIS Quarterly*, Vol. **23**(1), pp. 67-93.
- Laanti, M., Salo, O. and Abrahamsson, P. (2011) Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation, *Information and Software Technology*, Vol. **53**(3), pp. 276-290.
- Laplante, P. A. (2003) Stand and Deliver: Why I Hate Stand-Up Meetings *ACM Queue*, Vol. **1**(7), pp. 7-9.
- Larson, C., E. and LaFasto, F., M. (1989) *Teamwork: what must go right / what can go wrong*, Sage Publications, London.
- Law, A. and Charron, R. (2005), Effects of Agile Practices on Social Factors Proceedings of the 2005 workshop on Human and social factors of software engineering, St. Louis, Missouri, pp.
- Layman, L., Williams, L. and Cunningham, L. (2006) Motivations and measurements in an agile case study, *Journal of Systems Architecture*, Vol. **52**(11), pp. 654-667.
- Lebas, M. and Weigenstein, J. (1986) Management Control: The Roles of Rules, Markets and Culture, *Journal of Management Studies*, Vol. **23**(3), pp. 259-272.
- Lee, A. S. (1991) Integrating Positivist and Interpretive Approaches to Organizational Research, *Organization Science*, Vol. **2**(4), pp. 342-365.
- Lee, A. S. (1999) Rigour and Relevance in MIS Research: Beyond the Approach of Positivism Alone, *MIS Quarterly*, Vol. **23**(1), pp. 29-33.
- Lee, G. and Xia, W. (2010) Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data, *MIS Quarterly*, Vol. **34**(1), pp. 87-114.
- Lee, S. M. (1971) In *Academy of Management Journal*, Vol. 14 *Academy of Management*, pp. 213-226.
- Lillis, A. M. (1999) A framework for the analysis of interview data from multiple field research sites, *Accounting & Finance*, Vol. **39**(1), pp. 79-105.

- Locke, K. (2001) *Grounded Theory in Management Research*, SAGE Publications, London.
- Lycett, M. and Paul, R. J. (1999) Information systems development: A perspective on the challenge of evolutionary complexity, European Journal of Information Systems, Vol. **8**(2), pp. 127-135.
- Macintosh, N. B. (1994) *Management accounting and control systems: An organizational and behavioral approach*, Wiley, Chichester, England.
- Mahmood, M., A. (1987) System Development Methods - A Comparative Investigation, MIS Quarterly, Vol. **11**(3), pp. 293-311.
- Mak, B. L. and Sockel, H. (2001) A confirmatory factor analysis of IS employee motivation and retention, Information & Management, Vol. **38**(5), pp. 265-276.
- March, J. G. (1965) *Handbook of Organizations*, Rand McNally, Chicago, IL.
- March, J. G. and Simon, H. A. (1958) *Organizations*, Wiley & Sons, Inc., New York.
- Marks, A. and Lockyer, C. (2005) Debugging the system: the impact of dispersion on the identity of software team members, International Journal of Human Resource Management, Vol. **16**(2), pp. 219-237.
- Markus, M. L. and Benjamin, R. I. (1996) Change agency -The next IS frontier, MIS Quarterly, Vol. **20**(4), pp. 385-407.
- Maruping, L. M., Venkatesh, V. and Agarwal, R. (2009) A Control Theory Perspective on Agile Methodology Use and Changing User Requirements, Information Systems Research, Vol. **20**(3), pp. 377-399.
- Mayer, R. C., Davis, J. H. and Schoorman, F. D. (1995) An Integrative Model of Organizational Trust, Academy of Management Review, Vol. **20**(3), pp. 709-734.
- McCracken, D. D. and Jackson, M. A. (1982) Lifecycle Concept Considered Harmful, Software Engineering Notes, Vol. **7**(2), pp. 29-32.
- McEvoy, J. and Butler, T. (2009) The role of project management in ineffective decision making within Agile software development projects, European Journal of Information Systems, Vol. **18**(4), pp. 372-383.
- Merchant, K. (1985) *Control in Business Organizations*, Pitman Publishing, Boston.
- Merchant, K. A. and Otley, D. T. (2007) A Review of the Literature on Control and Accountability In *Handbook of Management Accounting Research Volume 2*, (Eds, Chapman, C. S., Hopwood, A. G. and Shields, M. D.) Elsevier, Oxford, England, pp. 785-802.
- Miles, M. B. and Huberman, M. A. (1994) *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd edition, Sage Publications, Thousand Oaks, CA.
- Mills, P. K. (1983) Self-Management: Its Control and Relationship to Other Organizational Properties, The Academy of Management Review, Vol. **8**(3), pp. 445-453.
- Mingers, J. (2000) The contribution of critical realism as an underpinning philosophy for OR/MS and systems, The Journal of the Operational Research Society, Vol. **51**(11), pp. 1256-1270.
- Mingers, J. (2001) Combining IS Research Methods: Towards a Pluralist Methodology Information Systems Research, Vol. **12**(3), pp. 240-259.
- Mingers, J. (2004) Real-izing information systems: critical realism as an underpinning philosophy for information systems, Information and Organization, Vol. **14**(2), pp. 87-103.
- Misra, S. C., Kumar, V. and Kumar, U. (2009) Identifying some important success factors in adopting agile software development practices, Journal of Systems and Software, Vol. **82**(11), pp. 1869-1890.
- Moe, N. B., Dingsøyr, T. and Dybå, T. (2010) A teamwork model for understanding an agile team: A case study of a Scrum project, Information and Software Technology, Vol. **52**(5), pp. 480-491.
- Moody, D. L. (2000), Building links between IS research and professional practice: Improving the relevance and impact of IS research, Proceedings of the twenty first International Conference on Information systems Brisbane, Queensland, Australia pp. 351-360

- Mowday, R. T., Porter, L. W. and Steers, R. M. (1982) *Employee-Organization Linkages: The Psychology of Commitment, Absenteeism and Turnover*, Academic Press; New York; London.
- Myers, M. D. (2009) *Qualitative Research in Business & Management*, Sage Publications, London, England.
- Myers, M. D. and Avison, D. (2002) *Qualitative Research in Information Systems*, Sage Publications, London, England.
- Myers, M. D. and Newman, M. (2007) The Qualitative Interview in IS research: Examining the Craft, *Information and Organization*, Vol. **17**(1), pp. 2-26.
- Nerur, S. and Balijepally, V. (2007) Theoretical Reflections on Agile Development Methodologies, *Communications of the ACM*, Vol. **50**(3), pp. 79-83.
- Nerur, S., Mahapatra, R. and Mangalara, G. (2005) Challenges of Migrating to Agile Methodologies, *Communications of the ACM*, Vol. **48**(5), pp. 72-78.
- Ngwenyama, O. K. and Lee, A. S. (1997) Communication richness in electronic mail: Critical social theory and the contextuality of meaning, *MIS Quarterly*, Vol. **21**(2), pp. 145-167.
- Nicholson, B. and Sahay, S. (2001) Some political and cultural issues in the globalisation of software development: Case experience from Britain and India, *Information and Organization*, Vol. **11**(1), pp. 25-43.
- Orlikowski, W. J. (1991) Integrated information environment or matrix of control? The contradictory implications of information technology, *Accounting, Management and Information Technologies*, Vol. **1**(1), pp. 9-42.
- Orlikowski, W. J. and Baroudi, J. J. (1991) Studying Information Technology in Organizations: Research Approaches and Assumptions, *Information Systems Research*, Vol. **2**(1), pp. 1-28.
- Otley, D., Broadbent, J. and Berry, A. (1995) Research in Management Control: An Overview of its Development, *British Journal of Management*, Vol. **6**(Special Issue), pp. 31-44.
- Otley, D. T. (1994) Management control in contemporary organizations: Towards a wider framework, *Management Accounting Research*, Vol. **5**(3-4), pp. 289-299.
- Otley, D. T. (1999) Performance Management: A framework for management control systems research, *Management Accounting Research*, Vol. **10**(4), pp. 363-382.
- Otley, D. T. and Berry, A. J. (1994) Case study research in management accounting and control, *Management Accounting Research*, Vol. **5**(1), pp. 45-65.
- Ouchi, W. G. (1977) The Relationship Between Organizational Structure and Organizational Control, *Administrative Science Quarterly*, Vol. **22**(1), pp. 95-113.
- Ouchi, W. G. (1978) Transmission of Control through Organizational Hierarchy, *Academy of Management Journal*, Vol. **21**(2), pp. 173-192.
- Ouchi, W. G. (1979) A Conceptual Framework for the Design of Organizational Control Mechanisms, *Management Science*, Vol. **25**(9), pp. 833-848.
- Ouchi, W. G. (1980) Markets, Bureaucracies, and Clans, *Administrative Science Quarterly*, Vol. **25**(1), pp. 129-141.
- Ouchi, W. G. and Maguire, M. A. (1975) Organizational Control: Two Functions, *Administrative Science Quarterly*, Vol. **20**(4), pp. 559-569.
- Ouchi, W. G. and Price, R. L. (1993) Hierarchies, Clans, and Theory Z: A New Perspective on Organizational Development, *Organizational Dynamics*, Vol. **21**(4), pp. 62-70.
- Päivärinta, T. (2001) The concept of genre within the critical approach to information systems development, *Information and Organization*, Vol. **11**(3), pp. 207-234.
- Palmer, S. R. and Felsing, J. M. (2002) *A Practical Guide to Feature-Driven Development*, Prentice Hall.
- Pan, G. S. C. (2005) Information systems project abandonment: a stakeholder analysis, *International Journal of Information Management*, Vol. **25**(2), pp. 173-184.
- Pare, G. (2004) Investigating Information Systems with Positivist Case Study Research, *Communications of AIS*, Vol. **2004** (13), pp. 233-264.

- Parnas, D. (2006) Agile Methods and GSD: The wrong solution to an old but real problem, Communications of the ACM, Vol. **49**(10), pp. 29-29.
- Pee, L. G., Tham, Z.-C., Kankanhalli, A. and Tan, G. W. (2008), Turnover in Information Systems Development Projects - Managing Forgetting, Pacific Asia Conference on Information Systems (PACIS 2008 Proceedings), pp. Paper 72.
- Pettigrew, A. M. (1990) Longitudinal Research on Change: Theory and Practice, Organization Science, Vol. **1**(3), pp. 267-292.
- Pino, F. J., Pedreira, O., GarcÃa, F., Luaces, M. R. and Piattini, M. (2010) Using Scrum to guide the execution of software process improvement in small organizations, Journal of Systems and Software, Vol. **83**(10), pp. 1662-1677.
- Polanyi, M. (1962) Tacit knowing: Its bearing on some problems of philosophy, Reviews of Modern Physics, Vol. **34**(4), pp. 601-616.
- Ramesh, B., Cao, L. and Mohan, K. (2006) Can Distributed Software Development be agile?, Communications of the ACM, Vol. **49**(10), pp. 41-46.
- Remenyi, D., Williams, B., Money, A. and Swartz, E. (1998) *Doing Research in Business and Management: An Introduction to Process and Method*, Sage Publications, London, UK.
- Rising, L. and Janoff, N. (2000) The Scrum Software Development Process for Small Teams, IEEE Software, Vol. **17**(4), pp. 26-32.
- Robey, D., Franz, C., R. and Farrow, D. (1989) Group process and conflict in systems development, Management Science, Vol. **35**(10), pp. 1172-1191.
- Robey, D. and Markus, M. L. (1984) Rituals in Information System Design, MIS Quarterly, Vol. **8**(1), pp. 5-15.
- Robinson, H. and Sharp, H. (2005), The Social Side of Technical Practices Proceedings of the Sixth International Conference on Extreme Programming and Agile Processes in Software Engineering (XP2005), Sheffield, UK 20-22 June, pp. 100-108
- Rousseau, D. M. (1998) Why workers still identify with organizations, Journal of Organizational Behavior, Vol. **19**(3), pp. 217.
- Royce, W. (1970), Managing the Development of Large Software Systems: Concepts and Techniques, Technical Papers of Western Electronic Show and Convention (WesCon), August 25-28, Los Angeles, USA, pp. 328-338
- Rustagi, S., King, W. R. and Kirsch, L. J. (2008) Predictors of Formal Control Usage in IT Outsourcing Partnerships, Information Systems Research, Vol. **19**(2), pp. 126-143.
- Schuh, P. (2004) *Integrating Agile Development in the Real World*, Charles River Media
- Schwaber, C., Leganza, G. and D'Silva, D. (2007) The Truth about Agile Processes: Frank Answers to Frequently Asked Questions, Forrester Report, Vol.
- Schwaber, K. (2004) *Agile Project Management with Scrum*, Microsoft Press, U.S.A.
- Schwaber, K. and Beedle, M. (2002) *Agile Software Development with Scrum*, Prentice Hall, NJ, USA.
- Sekaran, U. (2002) *Research methods for business: a skill building approach*, 4th, Wiley & Sons, New York , USA.
- Sfetsos, P. and Stamelos, I. (2010), Empirical Studies on Quality in Agile Practices: A Systematic Literature Review, 7th International Conference on the Quality of Information and Communications Technology (QUATIC), Porto, Portugal, September 29th-Oct 2nd, pp. 44-53
- Sheth, B. (2009), Scrum 911! Using Scrum to Overhaul a Support Organization, Agile Conference, 2009. AGILE '09., pp. 74-78
- Simons, R. (1995a) Control in an Age of Empowerment, Harvard Business Review, Vol. **73**(2), pp. 80-88.
- Simons, R. (1995b) *Levers of Control: How Managers Use Control Systems to Drive Strategic Renewal* Harvard Business School Press.
- Sliger, M. and Broderick, S. (2008) *The Software Project Manager's Bridge to Agility*, Addison Wesley.

- Snell, S. A. (1992) Control Theory in Strategic Human Resource Management: The Mediating Effect of Administrative Information, *Academy of Management Journal*, Vol. **35**(2), pp. 292-327.
- Soon, A. and Slaughter, S. (2004) Turnover of information technology professionals: the effects of internal labor market strategies, *ACM SIGMIS Database* Vol. **35**(3), pp. 11-27.
- Stake, R. E. (1994) Case Studies In *Handbook of Qualitative Research*, 2nd edition. (Eds, Denzin, N. K. and Lincoln, Y. S.) Thousand Oaks, CA, pp. 236-247.
- Subramaniam, V. (2006) *Practices of an Agile Developer: Working in the Real World*, Raleigh, N.C., Pragmatic Bookshelf.
- Sutherland, J. (2004) Agile Development: Lessons learned from the first Scrum, [Accessed Online] January 14th, 2011 <http://www.scrumalliance.org/resources/35>
- Sutherland, J., Viktorov, A., Blount, J. and Puntikov, N. (2007), Distributed Scrum: Agile Project Management with Outsourced Development Teams, *System Sciences*, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, pp. 274a-274a
- Takeuchi, H. and Nonaka, I. (1986) The New New Product Development Game, *Harvard Business Review*, Vol. (Jan/Feb), pp. 137-146.
- Tannenbaum, A. S. (1962) Control in Organizations: Individual Adjustment and Organizational Performance, *Administrative Science Quarterly*, Vol. **7**(2), pp. 236-257.
- Tannenbaum, A. S. (1968) *Control in Organizations*, McGraw-Hill, Inc., New York, USA.
- Tolfo, C. and Wazlawick, R. S. (2008) The influence of organizational culture on the adoption of extreme programming, *Journal of Systems and Software*, Vol. **81**(11), pp. 1955-1967.
- Towry, K. L. (2003) Control in a Teamwork Environment - The Impact of Social Ties on the Effectiveness of Mutual Monitoring Contracts, *Accounting Review*, Vol. **78**(4), pp. 1069-1095.
- Turner, K. L. and Makhija, M. V. (2006) The Role of Organizational Controls in Managing Knowledge, *Academy of Management Review*, Vol. **31**(1), pp. 197-217.
- van der Meer-Kooistra, J. and Scapens, R. W. (2008) The governance of lateral relations between and within organisations, *Management Accounting Research*, Vol. **19**(4), pp. 365-384.
- Van Der Vegt, G. S. and Bunderson, J. S. (2005) Learning and Performance in MultiDisciplinary Teams: The Importance of Collective Team Identification, *Academy of Management Journal*, Vol. **48**(3), pp. 532-547.
- VersionOne (2009) State of Agile Development Survey 2009 (Accessed: 31 March 2010) [Online]. Available at <http://pm.versionone.com/StateofAgileSurvey.html>
- Vlaanderen, K., Jansen, S., Brinkkemper, S. and Jaspers, E. (2011) The Agile Requirements Refinery: Applying SCRUM Principles to Software Product Management, *Information and Software Technology*, Vol. **53**(1), pp. 58-70.
- Walsham, G. (1995) The emergence of interpretivism in IS research, *Information Systems Research*, Vol. **6**(4), pp. 376-394.
- Walz, D., B. , Elam, J., J. and Curtis, B. (1993) Inside a software design team: Knowledge acquisition, sharing, and integration, *Communications of the ACM*, Vol. **36**(10), pp. 63-77.
- Weber, R. (2004) The Rhetoric of Positivism Versus Interpretivism, *MIS Quarterly*, Vol. **28**(1), pp. p3-xiii.
- Wenzel, M. (2004) Social Identification as a Determinant of Concerns about Individual-, Group-, and Inclusive-Level Justice, *Social Psychology Quarterly*, Vol. **67**(1), pp. 70-87.
- Westphal, J. D. and Khanna, P. (2003) Keeping Directors in Line: Social Distancing as a Control Mechanism in the Corporate Elite, *Administrative Science Quarterly*, Vol. **48**(3), pp. 361-398.

- Whitworth, E. and Biddle, R. (2007a), Motivation and Cohesion in Agile Teams, In Proceedings of the 8th International Conference, XP 2007, Como, Italy, June 18-22, pp. 62-69
- Whitworth, E. and Biddle, R. (2007b), The Social Nature of Agile Teams, Proceedings of AGILE 2007, Washington D.C, USA, pp. 26-36
- Wilkins, A. L. and Ouchi, W. G. (1983) Efficient Cultures: Exploring the Relationship Between Culture and Organizational Performance, *Administrative Science Quarterly*, Vol. **28**(3), pp. 468-481.
- Williams, L. (2003) The XP Programmer: The Few-Minutes Programmer, *IEEE Computer*, Vol. **20**(3), pp. 16-20.
- Williams, L. (2007) *A Survey of Agile Development Methodologies*
<http://agile.csc.ncsu.edu/SEMATERIALS/AgileMethods.pdf>
- Williams, L. and Cockburn, A. (2003) Agile software development: it's about feedback and change, *Computer*, Vol. **36**(6), pp. 39-43.
- Williams, L., Kessler, R. R., Cunningham, W. and Jeffries, R. (2000) Strengthening the case for pair programming, *Software, IEEE*, Vol. **17**(4), pp. 19-25.
- Wolcott, H. F. (1990) On seeking - and rejecting - validity in qualitative research In *Qualitative Inquiry in Education: The Continuing Debate*, (Eds, Eisner, E. W. and Peshkin, A.) Teachers College Press, New York, pp. 121-152.
- Wyatt, J., C. (2001) Management of Explicit and Tacit Knowledge, *Journal of the Royal Society of Medicine*, Vol. **94**(1), pp. 6-9.
- Xu, B., Lin, Z. and Xu, Y. (2011) A Study of Open Source Software Development from Control Perspective, *Journal of Database Management*, Vol. **22**(1), pp. 26-42.
- Yin, R. K. (1981) The Case Study Crisis: Some Answers, *Administrative Science Quarterly*, Vol. **26**(1), pp. 58-65.
- Yin, R. K. (2003) *Case Study Research: Design and methods*, 3rd, Sage Publications, Thousand Oaks, CA, USA
- Yin, R. K. (2009) *Case Study Research: Design and Methods*, 4th, Sage Publications, Thousand Oaks, CA, USA

Appendix A – Interview Protocol

This appendix details the interview protocol used in each of the interviews. Where used prompts are detailed in italics after each question.

SECTION A: RESEARCH OBJECTIVES

This study aims to investigate one single control mode, namely clan control, in ASD teams through the following objectives:

- (a) Develop a framework of the pre-conditions and resultant characteristics of clan control.
 - (b) Use this framework to investigate how agile practices help ASD teams to achieve the resultant characteristics of clan control.
 - (c) Develop a set of recommendations to identify how agile practices may be improved to achieve higher levels of clan control.
-

SECTION B: TABLE OF AGILE PRACTICES FOR REFERENCE

A copy of this table was provided to each participant for reference at any point during the interview.

Agile Practice	Description
Sprint Planning Meeting	The sprint planning session is a meeting that takes place at the start of each sprint where the team collectively define and plan tasks that must be completed during the next sprint.
Daily Scrum Meeting	The daily scrum is a short daily status team meeting lasting a maximum of 10-15 minutes typically conducted at the same time each day. The meeting is conducted with team members standing up. During the meeting team members explain briefly what they accomplished since the previous meeting, what will be completed by the next meeting and indicate any impediments that may prevent them from completing their current tasks
Sprint Review and Retrospective Meeting	A sprint review and retrospective is a meeting that is held at the end of each sprint where the project team demonstrates the software completed to the customer and to management to obtain feedback. The team reflect on what went well in the sprint, what did not, and what could be improved for future sprints.

SECTION C: BACKGROUND QUESTIONS

C.1 Personal Demographics

- (a) How many years experience do you have working in the software industry?
- (b) How many years have you been employed by this organisation?
- (c) How many years experience do you have in agile software development?
- (d) What is your role on this project?
[Looking for skill diversity]
- (e) How long have you been part of this team?
- (f) How would you describe your level of expertise/knowledge in relation to the agile methodology?

C.2 Team Background

- (a) How long has this project team been working together?
[To gain an understanding of whether the team have developed a common understanding, beliefs, expectations over a long period of time. How well are they socialised into the group?]
- (b) Has the team changed intermittently, or has it evolved over time?
- (c) How long has the team been using an agile methodology to develop software?
- (d) What roles are present in the team?
[Product Owner, Scrum Master, Testers, Project Manager, Developer]
- (e) Is the project team distributed?
[To identify different culture and /time zones that may impact on the project]
 - i. If yes, where are other team members located?
- (f) Is the customer co-located with the project team?

C.3 Project Background

- (a) Please provide some background information on the project you are currently working on
[Purpose, complexity, criticality, client, users or other relevant info]
- (b) What is the total duration of the current agile project in months/weeks?
- (c) How many sprints are planned for the current agile project?

- (d) Which sprint is the project currently working on?
- (e) What is the budget for the project?
- (f) Is it a fixed price project or time and materials?

C.4 Agile Methodology

- (a) Which agile methodology is used by the project team?
- (b) Is a customised version of the methodology used?

[Customised, hybrid with Waterfall or other method, implemented as is]

SECTION D: IDENTITY

- (a) Do you feel part of the team? Why/Why not?
 - i. If not, can you make any suggestions as to how this may be addressed?
 - ii. How do the 3 agile practices support/inhibit this?
- (b) Do you consider the customer to be part of the team?
- (c) How does the use of the agile practices impact on the relationship with the customer?

[e.g. requirements were better defined, communication was improved with the customer, made it more difficult]
- (d) How does co-location/team distribution impact on the use of the agile practices?
- (e) Are team members cooperative or competitive?
 - i. How do the agile practices inhibit/support this?
 - ii. Can you provide an example?
- (f) How are new team members integrated into the team?

[Are they recruited specifically for an agile team? How are they socialised into the group? E.g. provision of information on how things are done, formal/informal on the job training, ad-hoc meetings, social events, sharing of knowledge, attendance at meetings]

 - i. How do the agile practices support this?
- (g) How are new distributed team members integrated into the project team?

[Are they informed of certain behaviours that they must adhere to?]

 - i. How do the agile practices support this?
- (h) Do the agile practices encourage dysfunctional behaviour in the team?

[E.g. agreeing collectively to reduce estimates, working late because everyone else is]

- i. How do they do this? Can you give an example?
 - ii. Can you make any suggestions as to how this may be addressed?
- (i) Does the team have access to sufficient (physical) resources to allow them to work in a timely, proactive and efficient manner?
- [E.g. tools, meeting space, software]*
- i. Have the agile practices had any impact from this perspective?

SECTION E: COMMITMENT

- (a) Do you think this project team is willing to put in a great deal of effort beyond that normally expected in order to help the project be successful?
 - i. Why, why not?
- (b) How do the 3 agile practices encourage commitment to the team?
- (c) If you could move to another team that was doing similar work would you be happy to move?
 - i. Why, why not?
- (d) Would you say that the project team inspires you to do your best in terms of job performance?
 - i. Why, why not?
- (e) Do the 3 agile practices impact on your performance on the team?
 - i. How do they do support/inhibit/prevent this?
 - ii. Can you make any suggestions as to how any this may be improved?

SECTION F: LOYALTY

- (a) How do the 3 agile practices encourage loyalty within the team?

[e.g. belong to the team, identify with the team]

 - i. How do they support/inhibit this?
 - ii. Can you make any suggestions as to how this may be improved?
- (b) Do you feel a sense of pride in your work?

- i. How do the 3 agile practices contribute to that?
- (c) How would you describe the team spirit?
[e.g. Good team spirit, poor team spirit, cohesive team]
- i. How do the agile practices impact on team spirit?
 - ii. Can you make any suggestions as to how any this may be improved?
-

SECTION G: COMMON GOALS

- (a) What do you perceive as the goal of the team?
[e.g. deliver working software at the end of every sprint, meeting the deadline for the next release, developing a good relationship with the customer, so that the team will get follow-on work, deliver on time, within budget, of good quality and meet customer expectations]
- (b) How do the 3 agile practices help the team to set goals for the project?
[Are the goals set by management, by the team, by the customer? Are they pre-specified or do they evolve as the project progresses? Do the team have different internal goals e.g. to be the team that always delivers on time and within budget, to be the team that has the lowest number of defect rates, best customer satisfaction rating?]
- i. How do they do this/inhibit this?
[In terms of job satisfaction, productivity, improved/decreased performance of the team as a whole, higher quality product, higher staff motivation]
- (c) To what extent do the 3 agile practices cause conflict between the personal goals of team and the goals of management/customer?
- (d) Do the 3 agile practices encourage teams to set high/unrealistic goals for themselves?
- i. How do they do this?
 - ii. Can you describe a situation where this occurred?
- (e) Do the 3 agile practices encourage teams to set realistic goals for themselves?
- i. How do they do this?
 - ii. Can you describe a situation where this occurred?

(f) How do the 3 agile practices help the team to determine whether goals are met or not?

- i. How do they do this/inhibit this?

[How are they monitored? Are there metrics in place, specific targets that must be met, or other measurable criteria?]

(g) Do you believe the team is capable of delivering on its goals?

[Teams with strong belief in their capabilities set higher goals, develop strategies to achieve their goals and persist in the face of setbacks. Do team members believe their goals are not-related? Do individuals believe that their individual interests are best served by the complete immersion of each individual in the team?]

SECTION H: BEHAVIOURS

(a) Are there behaviours that are accepted/enforced by the team?

- i. Are these pre-specified, or have they evolved over time?
- ii. How have the 3 agile practices contributed to the development of these behaviours?

(b) Are there behaviours that are considered unacceptable by the team?

- i. Do you consider any behaviour of other team members unacceptable?
[E.g. arriving late, being late for meetings, lack of contribution, lack of attention to detail, failure to raise issues. Is it acceptable to stop by someone's desk for a conversation? Is it acceptable to discuss issues outside of meetings?]

(c) Do you monitor the activities of other team members?

[E.g. arriving late, being late for meetings, lack of contribution, failure to raise issues]

- i. How do the 3 agile practices support/inhibit this?

[Have they changed the behaviour of the team e.g. as a result of now having more meetings, more interaction, more communication?]

(d) Do all team members, including the customer, actively participate in team meetings?

- i. How do the 3 agile practices prevent/support this?

SECTION I: INFLUENCE

- (a) How do the 3 agile practices encourage team members to influence each other and each other's behaviour? These can be either positive or negative influences [e.g. *Daily scrum provides facility to gently remind team members that a particular piece of work needs to be completed, is persuasion used, is encouragement used, is advice/feedback provided to team members/management/customers?*]
- i. Does this occur on a regular basis?
 - ii. Can you give an example of this?
 - iii. Can you make any suggestions as to how any negative influences may be addressed?
- (b) Does peer pressure exist in the team?
- i. Is it used to exert control?
 - ii. Can you give an example of this?
- [This can be used in a positive or negative way – peer pressure/peer support, pressure during stand-ups to deliver]*
- iii. How do the 3 agile practices encourage/discourage the use of peer pressure?
 - iv. To what extent do you feel peer pressure influences/impacts on the project team?
 - v. Can you describe an instance where peer pressure was exerted by the team members on you as a project manager, or on another member of the team?
- (c) Can you detail any cultural issues that have influenced the behaviour of the team in another (distributed) location?
- [e.g. distributed teams in other locations worldwide where the values and beliefs of cultures are different and need to be accommodated]*
- (d) Can you make any suggestions as to how any negative influences may be addressed?

SECTION J: SHARED VALUES & BELIEFS

(a) How would you describe your current working environment?

[How would you describe the team morale? Do team members feel individually responsible for the groups' performance and help/teach other the skills require?]

- i. How have the 3 agile practices impacted on the working environment?
- ii. Have they made the working environment better or worse than before?

(b) What qualities, characteristics do you place most value in and why?

[e.g. leadership, technical ability, teamwork, ability to learn, willingness to learn, open communication, honesty and integrity, trust, accountability, quality, team/personal image, customer interaction?]

- i. How do the 3 agile practices contribute to/inhibit the development of these values?

[Provide additional opportunity for communication, provide transparency, provide public accountability, awareness of individual competencies]

(c) How do the 3 agile practices contribute to/inhibit the beliefs of the team?

SECTION K: EVALUATION, REWARDS AND SANCTIONS

(a) How are rewards distributed to the team?

[Are rewards distributed equally among team members? How is the contribution of each team member evaluated? What happens if one individual made a large contribution to the success of the project? Who distributes the rewards? When do team members receive rewards?]

(b) How do the 3 agile practices contribute to the determination of rewards for the team?

[Are rewards based on the performance of the team or on individual performances within the team?]

(c) How do the 3 agile practices help to reward/sanction the team?

- (d) What formal rewards are available to the project team on completion of a successful project?

[E.g. promotion, salary increase, bonus, social night out, ensure the team continues to work together, customer retains the team to complete further work]

- (e) What informal rewards are available to the project team on completion of a successful project?

[Informal night out at the end of a sprint/milestone/project, public recognition for a job well done]

- (f) Does the team informally reward/sanction themselves on completion of a successful sprint/project?

- i. How is this done?

[e.g. slack time before the next project commences, early finish, no overtime, longer lunches]

- (g) To what extent are team members motivated by monetary or non-monetary rewards?

- i. Can you describe an instance where the team was motivated by monetary or non-monetary rewards?

- (h) Do you reward/sanction yourself on the completion of a successful sprint/project?

- i. How is this done?

- (i) Are there any repercussions/sanctions for individuals/the project team if the sprint/project is not successful?

[Lack of bonus/monetary reward, reprimand, placed on lower priority projects]

- (j) How would you describe the performance of the team?

[Does the team perform well? Are they considered successful, adequate, or inadequate?]

Appendix B – Examples of Data Coding

This appendix details some examples of the data coded during the analysis phase of this study.

Quote	Source	Key Codes
Once a person “joins the team they are expected to contribute to the sprint planning piece.”	Developer 3, C1	Identity\Members feel part of the team\SPM
Team members “decide in each sprint what tasks [will be done] and who’s going to do them”	Developer 2, C1	Identity\Similar goals\SPM
“I have personally worked more overtime now since the start of Scrum”	Developer 7, C2	Commitment\Expend effort\SPM
“the sprint planning on Monday means that me as Product Owner has to work Sunday evening to be prepared Monday morning.”	Product Owner, C2	Commitment\Expend effort\SPM
“you understand the character of the different people on the team...you have a relationship with them as soon as possible and you are involved in what the team is doing”	Developer 5, C3	Identity\Members feel part of the team\DSM
“the continuous [daily] communication between the team, helped me feel part of the team and responsible for my tasks”	Technical Architect, C1	Identity\Members feel part of the team\DSM
It was expected that “we are on time for the daily scrum meetings”	Developer 3, C3	Shared norms\DSM
“It is a very good thing, to think back and analyse how we worked”	Developer 1, C2	Identity\Common goals\SRRM
“we determine how we can improve things [from a QA perspective]”	QA, C3	Identity\Common goals\SRRM
“the retrospective provides you with feedback or praise that you have done a good job”	Developer 7, C2	Reward\SRRM

B -1 Examples of Coding by Resultant Characteristic