2016

Understanding Agile Software Development Assimilation Beyond Acceptance

Corey Wayne Baham
Louisiana State University and Agricultural and Mechanical College

UNDERSTANDING AGILE SOFTWARE DEVELOPMENT ASSIMILATION BEYOND ACCEPTANCE

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Information Systems and Decision Sciences

by
Corey W. Baham
B.S., Louisiana State University, 2004
M.S., Southern University and A&M College, 2011
May 2016

ACKNOWLEDGMENTS

TABLE OF CONTENTS

ABSTRACT

Agile software development methods represent a departure from the heavily regimented and document-driven procedures of traditional, waterfall approaches. Despite the highly touted benefits of employing agile ISD methods and the growth of agile adoption rates over the past two decades, it is not clear why some organizations fail to routinize agile methods, while others do so and realize their promised benefits. Motivated by the need to understand the factors that influence agile routinization, this study empirically examines the deep contextual factors that impact the extent to which agile methods are proliferated throughout an organization. Findings indicate that project success from initial agile use does not translate to routine agile use. Instead, findings from the study suggest that organizational factors of organizational culture and structure play a pivotal role in the routinization of agile methods.

CHAPTER 1: INTRODUCTION

This dissertation titled, "Understanding Agile Software Development Assimilation Beyond Acceptance," which purposes to investigate and broaden the understanding of routinization and infusion of agile software development in organizations in the post-adoption phases. In this research, I develop and test a new model for the routinization of agile software methods in organizations. The first essay assesses the theoretical perspectives that influence our understanding of agile software development, offers insights on the application of these perspectives, and provides guidance for future studies. The second essay, a comparative case study on agile assimilation, builds on the first essay by examining the contextual factors of post-adoptive agile use through the lens of diffusion of innovation's theory. The third essay, a research design for a future field study, includes the factors that were proposed in the first essay and validated in the second essay.

CHAPTER 2: TOWARD A THEORY OF AGILE SOFTWARE DEVELOPMENT ROUTINIZATION

INTRODUCTION

During the 1990s, the software development practice underwent unprecedented changes due to the emergence of object-oriented programming and the widespread use of the Internet. Increases in production speed, efficiency, and agility became vital for firms that were seeking to compete in a more globally connected economy. The pressure to be early to market challenged software development teams to develop faster, more agile processes in order to produce more frequent iterations of working software. Developers began to view the document-driven and heavily regimented procedures of traditional approaches as inadequate impediments to their ability to respond to user requirement changes and to collaborate with customers. In 2001, a group of practitioners gathered to discuss the shortcomings of heavyweight methods in an attempt to unite around common software development principles. The result was the *Manifesto for Agile Software Development*, which communicated their chief values and popularized the term "agile" (see Appendix A1; Beck et al. 2001). By the early 2000s, several agile software development (ASD) methods were created, including: eXtreme Programming, Scrum, Dynamic Systems Development Method, Adaptive Software Development, Crystal, Feature-driven Development, and Pragmatic Programming. Furthermore, ASD adoption rates in organizations continued to increase as more practitioners turned to lightweight, ASD methods. In a 2007 survey, 69% of respondents indicated that their organizations were using ASD, and 85% of organizations using ASD methods had completed more than one

project, which suggested that ASD had gone beyond the pilot project stage (Ambler 2007). Recent surveys indicate that ASD adoption continues to burgeon (Version One 2015), as organizations make ASD the principal method for developing software.

Despite the widespread adoption of ASD methods in practitioner communities, the academic research community as a whole has been slow to understand the phenomenon with theoretical underpinnings and empirical support (Dybå and Dingsøyr 2008). As a nascent research stream, early ASD research was criticized for its lack of rigor (Abrahamsson et al. 2009), its lack of originality as a software methodology (Hikka et al. 2005), and its lack of empirical results (Mcbreen 2003). Information systems (IS) journals, in particular, published fewer studies on ASD than computer science (CMPS) and software engineering (SE) journals did (Dybå and Dingsøyr 2008, Dingsøyr et al. 2012). ASD researchers responded to these concerns by increasing the number of empirical studies, including hundreds of peer-reviewed publications, and focusing on the development and application of theory. The growing body of ASD research led to two special issues in 2009 by IS journals, *Information Systems Research* and *European Journal of Information Systems,* and in 2012 by SE journal, *Journal of Software and Systems*. In addition, a number of introductions, overviews, and systematic literature reviews of ASD methods were published (Cohen et al. 2004; Erickson et al. 2005; Ågerfalk et al. 2009; Abrahamsson et al. 2009).

Despite some important contributions to the ASD literature, there remain concerns that ASD researchers should address as the research stream continues to mature. First, in the

attempt to provide greater theoretical support, scholars adopted various theoretical lenses to understand aspects of ASD, however, scholars' varied interests in ASD research contributed to a fragmented view of the phenomenon (Cao 2004; Conboy and Duarte 2010; Wang et al. 2012; Ghobadi and Mathiassen 2015). There is a still a need for more unified theoretical understanding of the ASD, starting with an analysis of the current body of literature. The present study addresses this issue by providing a rich synthesis of the theoretical contributions of ASD research. Second, knowledge fragmentation is further amplified by the lack of a current and comprehensive review of the theoretical contributions to ASD research. Such a review is necessary to understand what is known and what is not known.

Prior systematic reviews describe various flavors of ASD methods, and identify common research themes and publication trends in ASD research (Dybå and Dingsøyr 2008; Dingsøyr et al. 2012; Senapathi and Srinivasan 2013; Hummel 2014). Additionally, both prior systematic reviews and special issues make recommendations for future research such as calls for more theoretical support in ASD research, more studies on post-adoptive ASD use, and a unified framework for understanding ASD methods (Dybå and Dingsøyr 2008; Abrahamsson et al. 2009; Dingsøyr et al. 2012), all of which are focuses of this review along with the inclusion of the IS perspective. To illustrate, Dybå and Dingsøyr's 2008 systematic review of ASD methods examined articles and conference proceedings that were published between 2001 and 2005. Although their work is rigorous, systematic, and well-cited, their 2008 review primarily covered ASD research in the computer science and software engineering disciplines mostly because the majority of IS studies on

ASD were published after 2005 (see Figure 2.1). The need for more IS representation is further realized in Dingsøyr et al.'s 2012 review on ASD research. IS journals, including *Management Information Systems Quarterly, Information Systems Research, Journal of Management Information Systems, and Decision Support Systems*, were not included in the journal search parameters, yet these journals published research on ASD between January of 2003 and November of 2014.[1]
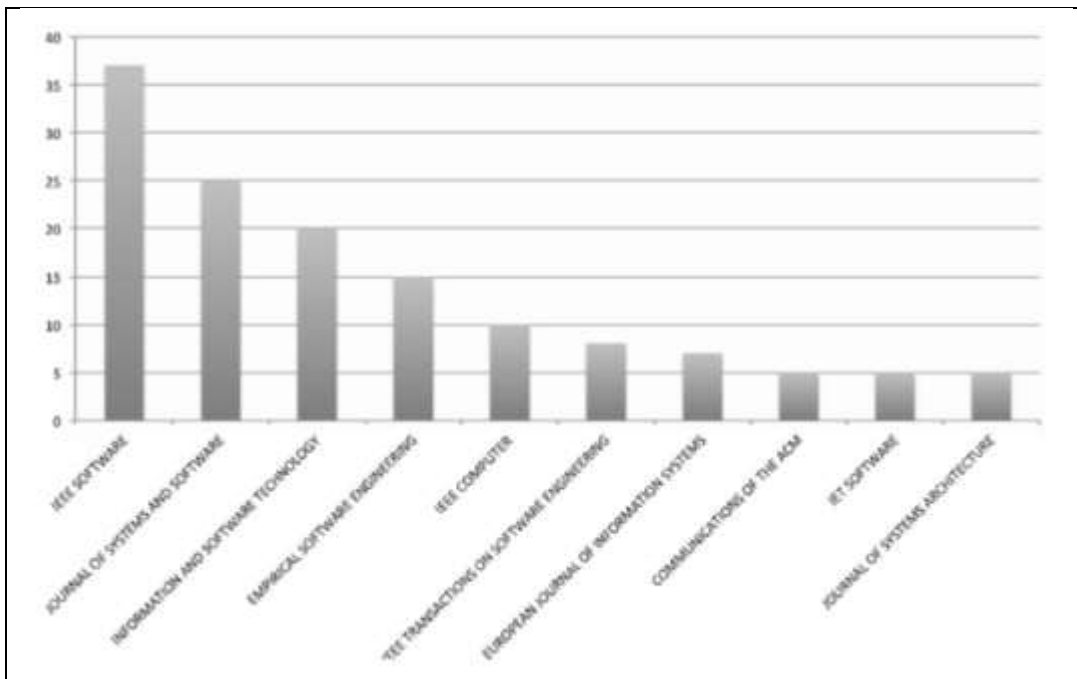
Figure 2.1. Agile papers in scientific journals in 2012 (From Dingsøyr et al. 2012)

Despite some shared publication real estate, the IS research community still differs from the software engineering research community in the sense that the IS community takes into account more of the social and organizational aspects of software development (Dhillon 1997; Baskerville 1998). Therefore, we believe that a more comprehensive

---

[1] Dingsøyr et al. 2012 did mention both EJIS's and ISR's special issue on agile in a separate section of their review.

review on the theoretical perspectives of ASD from both software engineering and IS research can help provide a more complete perspective of the social and technical aspects of ASD methods in organizations. This research is motivated by our desire to extend extant literature by providing an understanding of ASD in organizations from a theoretical perspective. We contend that the development and application of theories are more capable in explaining the nature and relationships that impact the ASD phenomenon than descriptive studies and lessons learned (Robey et al. 2008). Nevertheless, we believe that the common themes, application of theoretical perspectives, and significant findings of prior ASD research are raw material for ASD theory development.

Moreover, the objective of this review is to build upon prior ASD research to provide a rich synthesis, which is a compendium of what is known versus the gaps in our knowledge, and to provide substantial guidance for future studies in the form of theory and propositions for the benefit of both research and practice. Our review will assess the theoretical perspectives that have been used to study ASD, provide insights on the application of these perspectives, and build upon the insights of the review to inform new theory (Leidner and Kayworth 2006). In doing so, the main contribution of this study is a novel model that conceptualizes the relationships between acceptance and routinization in the ASD assimilation process. In addition, our model can help practitioners to understand the organizational issues that influence the ASD routinization process as well as help them understand some of the misnomers concerning the role of project success factors in ASD routinization. Key terms associated with our model are defined in Appendix A2.

The researchers fulfill this purpose by employing a rigorous search methodology as outlined in prior systematic reviews and systematic review guides for searching the literature (Dybå and Dingsøyr 2008, Dingsøyr et al. 2012; Kitchenham and Charters 2007; Fink 2005; Okoli 2010; Ridley 2008; Webster and Watson 2002). This review uses instructions, search the literature, identify specific inclusion and exclusion criteria, test for quality, extract data, synthesize studies, and write a review (Okoli 2010). This work is exploratory in its data gathering process, which is used in theory building. The researchers use the eight-step procedure to make the following contributions. First, we review the ASD literature and identify existing theoretical perspectives in ASD research. Second, we synthesize the theoretical contributions across key themes in the ASD literature. Third, we introduce an integrated framework for understanding the relationships among the themes in ASD research. Fourth, we develop a second framework that integrates the level of analysis with stages in the ASD assimilation process in order to provide further insight into the research gaps from an assimilation perspective. Finally, we propose a novel model on ASD routinization and conclude by highlighting areas for future research.

Our research questions served as a roadmap that we followed during the literature search and analysis. In order to assess the theoretical perspectives that influence our understanding of ASD within the IS field, this study investigated the following research questions (RQ):

RQ1. What are the theoretical perspectives in ASD research?

RQ2. What insights have theory-testing/theory building approaches provided on ASD methods?

RQ3. Where are the gaps in our theoretical understanding of ASD research?

RQ4. How can insights from theory-testing/theory building approaches be expanded in future research?

The remainder of the article proceeds as follows. First, the research methodology section will lay out the search strategy and screening process used to comb through the literature. Second, the results section will identify common themes found during the literature search and provide insights of theory building and theory testing approaches in ASD research. Third, we elaborate on the research findings by highlighting knowledge gaps. Fourth, the discussion section concludes this work with the presentation of a new model and considerations for future research.

RESEARCH METHODOLOGY

Our literature review necessitated a protocol (or plan that describes the conduct of the review), which was developed by following existing guides and procedures for conducting systematic literature reviews (Kitchenham and Charters 2007; Okoli 2010, Webster and Watson 2002) and by following examples from published reviews (Dybå and Dingsøyr 2008, Dingsøyr et al. 2012; Abrahamsson et al. 2002, 2009). Our research protocol includes the development of the reviewer training manual (see Appendix A3). Each step is detailed for replication in future studies, and by specifying the search criteria in the protocol beforehand, the reviewers were able to minimize the effects of a selection

bias. The training manual was used to ensure procedural consistency in the execution of this study among its reviewers. Experts in the IS field were consulted concerning the search methodology so as to ensure completeness of the literature search before moving onto the analysis[2] (Petticrew and Roberts 2006, Fink 2005, Okoli 2010).

Search Strategy

The search strategy consists of finding theoretical perspectives in studies on ASD. Although theoretical perspectives from conceptual papers are included in our analysis, our final inclusion criteria is restricted to empirical studies because (1) the lack of and need for more empirical justification for the ASD phenomenon has been clearly stated in prior research (Abrahamsson et al. 2009; Dybå and Dingsøyr 2008), and (2) prior reviewers were persuaded that empirical studies best demonstrate the influence of theoretical choices on the formulation of research findings (Robey et al. 2008). In Robey et al.'s (2008) review, they stated, "theories presented in non-empirical papers have not yet demonstrated their influence over research practice." Thus, empirical studies were the most appropriate choice for fulfilling our intention of building a theoretical understanding of our phenomenon. Given the enormity of studies on ASD, we chose to limit the initial sample to ASD studies published in IS and related journals that have been recognized in previous IS journal quality assessments (Rainer and Miller 2005; Lowry et al. 2004; Katerat-tanakul et al. 2003; Peffers and Tang 2003; Mylono-poulos and Theo-harakis 2001; Whitman et al 1999). These journals are listed on the Association for Information Systems' webpage, titled "MIS Journal Ranking" (2014). This strategy was implemented

---

[2] Experts included Joey George and Juhani Iivari.

in order to prevent having an unmanageable number of articles that yielded little value (Leidner and Kayworth 2006).

This review was grounded in articles published between January 2001 and November 2014 in IS and related journals. The main search algorithm limited results by using the phrases, "agile," "software development," "software methodology," "theory," "theoretical," "assumptions," and the necessary Boolean operators[3] using Business Source Complete, an EBSCO search engine. Afterwards, additional steps were taken to ensure that the search strategy was comprehensive. First, each of the 110 journals on the Association for Information Systems' MIS Journal Ranking list including the special issues from ISR (#2), EJIS (#11), and JS&S (#70) was manually searched to ensure comprehensiveness (see Appendix A4). Next, the 2013 and 2014 conference proceedings of HICCS, AMCIS, and ICIS were searched. Finally, the researcher searched the bibliography of previous systematic literature reviews to ensure that additional articles, particularly those in software engineering and other relevant journals, had not been overlooked. Altogether, this process yielded 154 articles from 40 different journals for our practical screen. Figure 2.2 shows a diagram of the number of papers that were excluded during each stage of the screening process. After each screening question, the number of studies that were excluded during each stage was inserted into the box on the right of the question. For instance, 50 studies were excluded after question 1 (Q1) and

---

[3] The Boolean operators "AND" and "OR" were used in order to limit results to articles that contained the word "agile" with the words "software methodology" or "software development," as well as the term "agile" with the words "theory," "assumptions," or "theoretical." Without the above constraints, a search yielded excluded studies, generated important studies that the researcher already knew, or contained a number of studies that was too large to manage. The following search terms and operators were used: ((agile software methodology) OR (agile software development)) AND (theory OR assumptions OR theoretical).

entered into the box labeled P.Q1, while the remaining 107* studies were passed onto the next screening question.

* Originally 50 studies were discarded after Q1. Three of these studies fullfilled the remaining criteria and were to the total number of studies after Q1 (104+3 = 107).
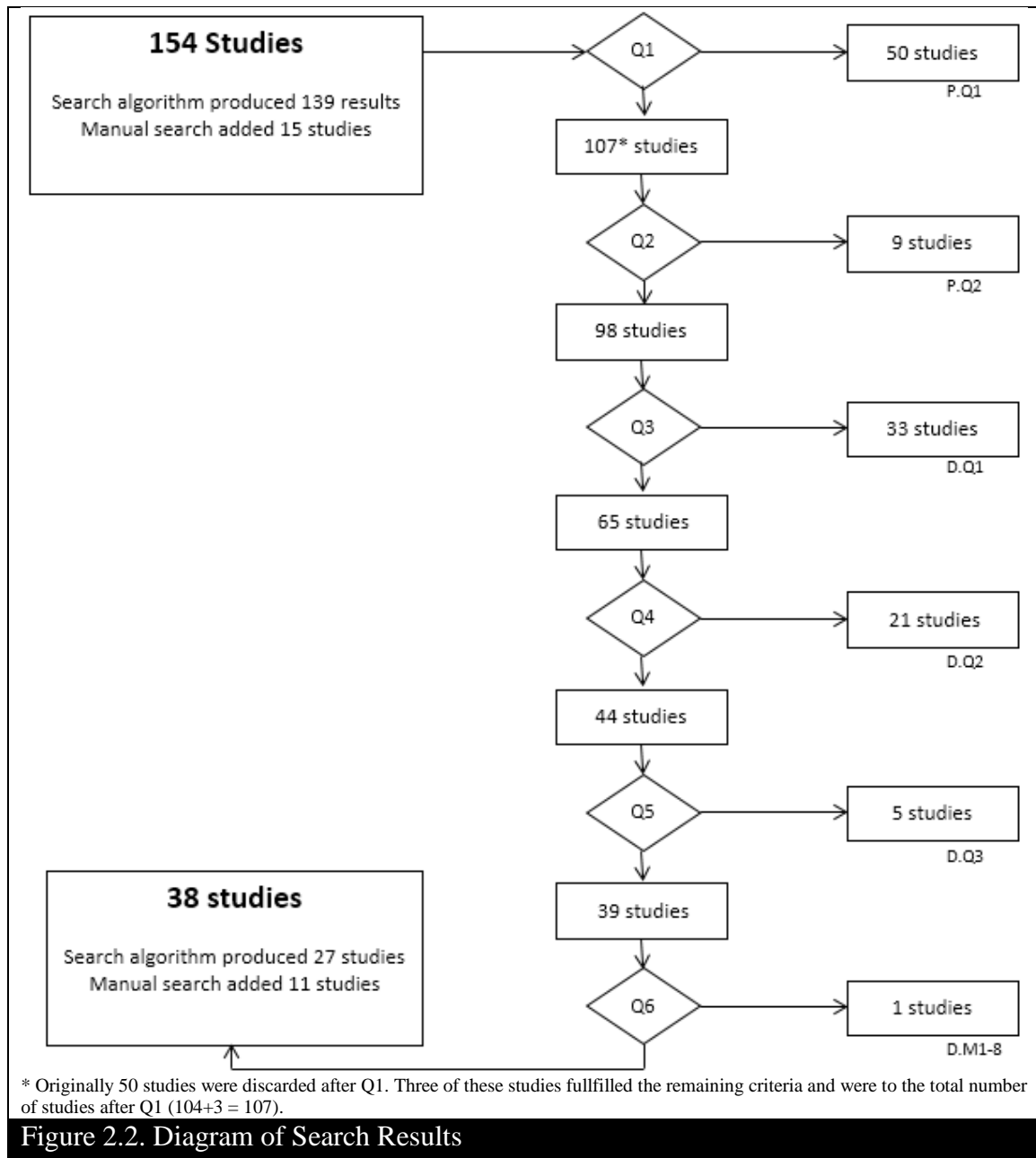
Figure 2.2. Diagram of Search Results

Practical Screen - Criteria for Inclusion and Exclusion of Studies in the Review

A total of three reviewers used the training manual (Appendix A3) to examine each article included in this study, which consisted of two reviewers for selecting articles for

inclusion and a third reviewer who acted as the "gold standard," or the deciding vote, if a disagreement occurred that would prevent the study from moving forward.

Two reviewers applied the practical screen to the resultant articles of the literature search in order to identify quickly articles that did not fit the basic qualification criteria for this study (see Appendix A3). First, articles were excluded if they were not published in an IS journal listed on the Association for Information Systems' webpage, titled "MIS Journal Ranking" (2014). Second, articles were not retained if they clearly did not pertain to ASD, as indicated by key search terms in the title, abstract, or keywords. Both quantitative and qualitative IS articles were included. Magazines, trade publications, and any other literature outside of peer-reviewed IS journals were also excluded from our analysis. Because this review's focus was on theoretical perspectives, papers that primarily contained lessons learned, opinions, tips, and strategies were excluded.

We developed and validated our preliminary screen by having two reviewers conduct a pilot test on five randomly selected articles. The screen was revised until the questions were clear and the results were consistent between the reviewers. Next, the reviewers applied the practical screen to each of the 154 articles. Finally, inter-rater agreement rate of 88% was calculated based on the number of observed agreements (n = 136). A kappa coefficient of .805 was calculated by dividing the observed agreement by the agreement possible beyond chance. A kappa coefficient of .805 is considered "almost perfect" (Fink 2005). Acceptable reliability scores range from 0.6-1.0 (Fink 2005). All disagreements were discussed and reconciled without needing the intervention of a third reviewer to act

as a gold standard. Only a few minor changes were added to the practical screen in order to distinguish ASD from ASD project management in other fields. After the final revision of our practical screen was executed, 50 (i.e. 47*) articles (see Figure 2.2) were removed after Q1 and 9 additional articles after Q2, leaving 98 studies for the detailed screen. Appendix A5 lists all 154 articles and the screen in which there are eliminated. The following subsections expound on the elimination decisions during each screening question and examine each cluster to ensure that helpful insights are not overlooked. We believe these sections show the thoroughness of our search strategy.

Practical Screen – Q1 (P.Q1)

Since there exists a myriad of academic journals with varying levels of quality, we used the AIS's journal ranking list, which is widely accepted by the IS field and includes peer-reviewed publications with metrics for measuring quality. The inclusion of over 100 journals is a very low acceptance criteria when considering that the IS field has an established basket of eight journals that are widely accepted for their academic rigor and review quality. This screen initially eliminates a total of 50 (i.e. 47*) studies. Our search algorithm yielded 23 studies from journals that either were outside of the AIS's journal ranking list, or were associated with other fields of study such as economics and quality assurance. Only one study was from a journal that is even remotely related to the field of IS. Our manual search through the SE literature yielded additional 27 studies, of which only three (Pikkarainen et al. 2008; Overhage and Schlauderer 2012; Senapathi and Srinivasan 2014) met our remaining screening criteria, which is outlined in Appendix A3. Thus, these studies are added to the totals of the subsequent screens.

Practical Screen – Q2 (P.Q2)

The next screen focused on eliminating studies that did not pertain to "agile" as a method of developing software. Although the nine studies removed during the practical screen included the phrase "agile software development" in the title, keywords, or the abstract, the use of ASD methods was not the central focus of the studies. Upon further examination, only Keith et al.'s (2013) contained a theoretical perspective, and is neither empirical nor holistic in its approach to studying agile methods. Keith et al.'s (2013) studied the application of coordination theory in service-oriented methodology for systems development.

Detailed Screen

Area 1 (Q3-Q5)

Once the initial screen was completed, the reviewers applied a detailed screen to each of the eligible articles. This screen consisted of two sections, in which the reviewers examined the body of each article, not just the title, abstract, and keywords. Since our goal is theory building, we chose to restrict our criteria to identify high quality articles, which can be used to development a model to guide future research. Therefore, the first section's, Area 1, exclusion criteria is as follows. First, reviewers excluded articles that did not provide a holistic perspective (i.e. too narrow, focusing on a single technique or practice, such as user stories, unit testing, and release planning) for understanding ASD. Prior literature reviews placed emphasis on articles that examined ASD holistically (Dybå and Dingsøyr 2008). Second, articles were excluded if they lacked empirical data or third, did not use a new or existing theory, or a theoretical lens to explain ASD. Past

research calls for more empirical and theoretical support in ASD research. Although our screening criterion excludes a number of articles, we provide a separate, yet detailed examination of the articles at each step to ensure that important theoretical contributions are not missed. The reviewers underwent a similar process for validating the detailed screen as they did for the practical screen. Five articles were selected and screened, and the results were compared until both reviewers were comfortable with the clarity of each question and the consistency of the results.

After two rounds of revisions to the Area 1 screening criteria, the instrument was reapplied to the 98 remaining articles. During this step, reviewers 1 and 2 each screened 49 articles, and their decisions were compared with that of the third reviewer. A kappa coefficient of .8930 was calculated to determine the inter-rater reliability of their inclusion and exclusion decisions. All disagreements were discussed and reconciled without requiring the intervention of a gold standard. After the revised Area 1 screen was applied, 59 of the 98 remaining articles were removed, leaving 39 articles that were eligible for the Area 2 screen. The following subsections expound on the screening decisions made during Q3-Q5 of the Area 1 screen.

Holistic: Thirty-three studies were removed because they did not contain a holistic view of ASD. Studies were considered holistic if they used focused on ASD in a broad sense as opposed to a focusing on a single technique. Holistic approaches to the study of ASD methods included topics such as ASD adoption, ASD usage, and ASD implementation. For example, Maruping et al. (2009a) used control theory to understand variables that

moderate the impact of ASD use on software project quality. In their study, both project outcome controls and informal, self-regulating controls moderated the relationship between ASD methodology use (holistic) and software project quality in the face of requirement changes. On the other hand, non-holistic studies were identified by the following approaches: 1) studies that focus on the single ASD techniques, or 2) studies that examine ASD methods to a degree, but do not place the study of ASD methods as its central focus. In the latter category, six articles briefly mentioned agile as one of many software development methods. These articles included topics on ambidexterity, tailoring, ecosystem, software platform strategies, and scoping. In the former category, seven articles study the XP practice of pair programming, while eight articles study the requirements gathering process. Other ASD techniques included continuous integration (2), test-driven development (2), user stories, release planning, and planning poker. The remaining studies consisted of a variety of topics from team member personality profiles to software engineering optimization techniques.

In this paper, we refer to a theoretical assessment as the use of at least one theoretical perspective within a single study. A theoretical perspective is referred to as an existing theory (i.e. control theory), an emergent theory, or a theoretical model or framework (i.e. TAM) used to examine ASD. Some studies such Harris, Collins, and Hevner (2009a) use multiple theoretical perspectives. Table 2.1 provides a distribution of the articles during the Area 1 screen. Overall, studies that contain a holistic view of ASD make up a higher percent of theoretical assessments, 74% versus 26%. Conceptual studies with a holistic in comparison to those that were conceptual, non-holistic, and theoretical (7%). In fact, only

10 of the 65 (15.3%) studies categorized as holistic failed to make a theoretical contribution.

Table 2.1. Distribution of Articles in Area 1							
	N, Q3	Empirical	Theory	Total	% Total	% Theory	% No Theory
Holistic							
	65	Y	Y	39	60%	85%	15%
		Y	N	5	8%		
		N	Y	16	25%		
		N	N	5	8%		
Non-holistic							
	33	Y	Y	14	42%	56%	44%
		Y	N	9	27%		
		N	Y	5	15%		
		N	N	5	15%		

Empirical: Next, twenty-one empirical studies were removed because they did not contain empirical data. Empirical studies accounted for a greatest percentage of theoretical assessments, 60% to 25% in holistic studies and 42% to 15% in non-holistic studies. In addition, only 7% of the theoretical assessments in this screen were found in studies that lacked both empirical data and a holistic perspective of ASD. Interestingly, holistic studies were comparatively equal or higher than non-holistic studies in every statistic except in empirical studies without theoretical assessments, 5 to 9. Taken together, prior research suggests that the use of both holistic approaches and empirical research have made more contributions to theory in ASD research in terms of the number of theoretical assessments. In fact, of the 21 theoretical perspectives found in the conceptual studies in this screen, only ten of them were later employed in empirical studies, which suggest that there remain opportunities for future empirical research. Theory: Next, five studies of the remaining 44 - holistic and empirical - studies were

removed because they did not contain a theoretical assessment of ASD. We searched for SE articles by extending the search algorithm found in the most recent systematic review on ASD methods (Dingsøyr et al. 2012). A total of 34 theoretical perspectives were found by extending the SE search algorithm compared to the 59 theoretical perspectives on ASD found using our search procedure in the IS literature (see Appendix A6). Four publications were found by both search procedures (Dingsøyr and Hanssen 2002; McAvoy and Butler 2007; Cao et al. 2009; Chan and Thong 2009) and one IS publication contained six theoretical perspectives (Batra et al. 2011). Additionally, a total of 68 out of the 98 theoretical assessments on ASD either were published in the AIS journal lists or selected IS conferences. Out of the sources that were found outside of the AIS list, only three articles (Pikkarainen et al. 2008; Overhage and Schlauderer 2012; Senapathi and Srinivasan 2014) passed the screening process. Furthermore, of our final 38 studies, there were 22 IS studies that passed the screening process compared to 16 SE studies passing the screening process. We now move to Area 2 of the detailed screen.

Area 2 (Q6)

Area 2 of the detailed screen applied a test for methodological rigor to each of the remaining articles. The researcher modified the quality criteria of the Critical Appraisal Skills Programme, which was used in a previous ASD review (Dybå and Dingsøyr 2008; C.A.S.P. 2012). Although it was not our intent in this study to reapply the peer-reviewing process, ensuring that each article contained a basic level of rigor was vital to our goal of producing a solid framework on ASD. Therefore, each article was examined for the presence or absence of basic research study components (Dennis and Valacich 2001). For

example, the reviewers asked, "Is the research design clearly specified?" and "Is there a clear statement of findings?" Appendix A3 contains the entire detailed screen criteria used in this study.

Area 2 of the detailed screen was performed in its entirety for each of the 39 articles. One point was assigned for a response of "yes," to each of eight questions and no points were assigned for a response of "no." Articles that failed to score five or more total points were excluded from the data extraction portion of the review. The authors chose a 5-point threshold because a score of 4 or less points would suggests that an article failed to make 50% or more of the fundamental aspects of a scientific study. The results yielded 38 final articles, as only one was eliminated due to an inadequate sample size.

Figure 2.3 shows the distribution of theoretical perspectives across the journals and select conference proceedings of the remaining 38 articles. A detailed description of the screening criteria for each paper is included in Appendix A3.
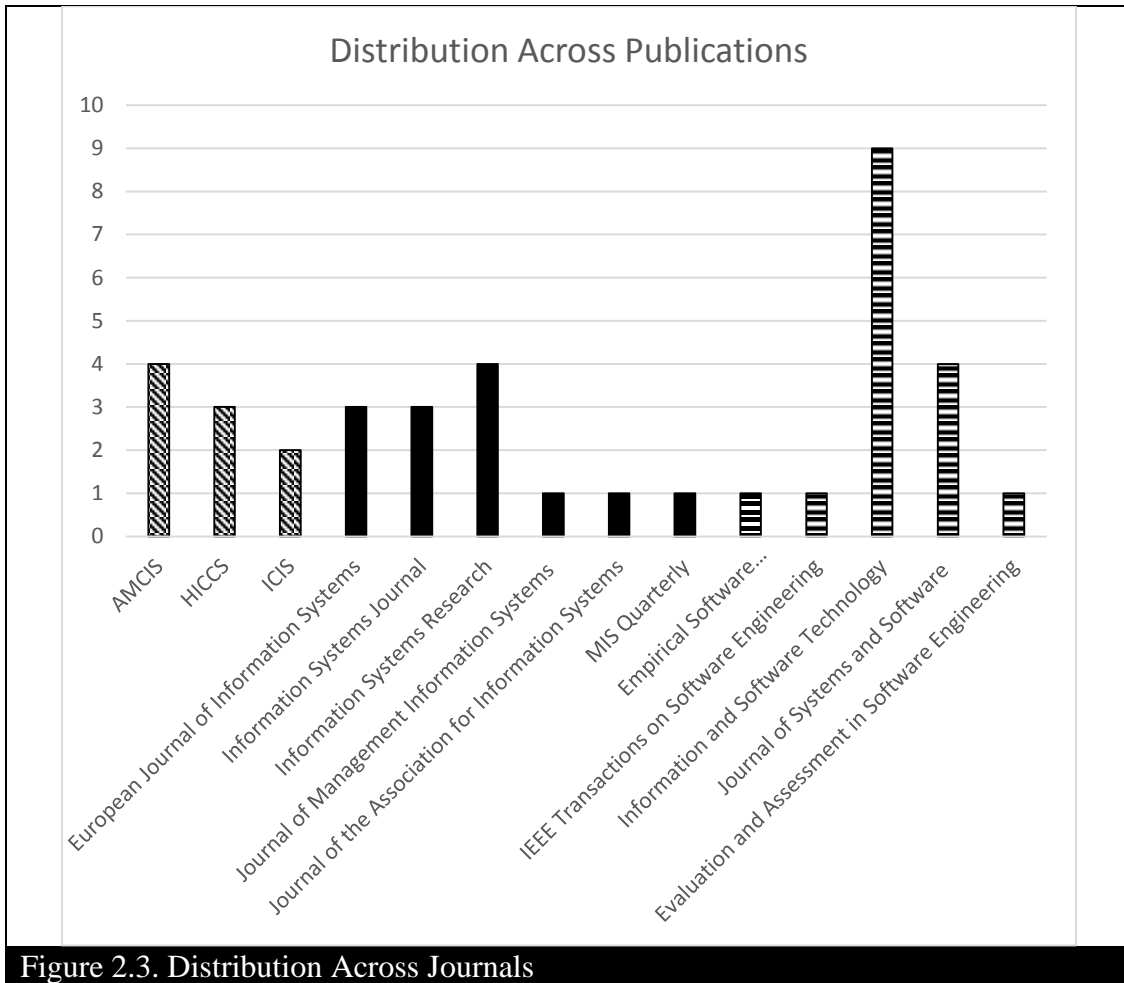
Figure 2.3. Distribution Across Journals

Data Extraction

During the screening process, the lead researcher extracted data from the articles using a general extraction form on articles that passed the practical screen and a more detailed extraction on articles that passed both the practical and detailed screens. Extracting data provides the researcher with a systematic way to answer questions and to record answers (Okoli 2010). Articles that passed the practical screen were extracted for the presence or absence of empirical data, a holistic view of ASD, and a theoretical assessment of ASD methods. The focus of each theoretical assessment was also recorded during the data

extraction stage. The detailed data extraction form was applied to the articles that passed the screening process as described in Appendix A7.

RESULTS

Descriptive statistics and general findings

After the general data extraction was complete, we were able to answer the first research question (RQ1), "*What are the theoretical perspectives in ASD research in the IS field?*" (see Appendix A6). A total of 90 of the 98 studies that remained after Q2 contained a theoretical assessment and maintained ASD as the primary focus (see Table 2.2). Of the 90 theoretical assessments on ASD, there were 61 unique theoretical perspectives from 40 different publication outlets.

Table 2.2. Agile Software Development Themes by Study Types		
Theme	Empirical	Conceptual
Introduction and Assimilation of ASD Methods	Conboy, Pikkarainen, and Wang (2007) Karlsson and Ågerfalk (2009) Cao, Peng, and Ramesh (2009) Mangalaraj, Mahapatra, Nerur (2009) Sarker and Sarker (2009) Drury, Conboy, and Power (2012) Overhage and Schlauderer (2012) Vijayasarathy and Turk (2012) Wang, Pikkarainen, and Conboy (2012) Russo, Shams, and Fitzgerald (2013) Senapathi and Srinivasan (2012; 2014)	Hazzan and Dubinsky (2005) Qumer and Henderson-Sellers (2008) Austin and Devin (2009) Chan and Thong (2009) Barlow et al. (2011) Senapathi, Drury, and Srinivasan (2013) Eck, Uebernickel, and Brenner (2014)

(Table 2.2 continued)

Theme	Empirical	Conceptual
Organizational factors and Governance in ASD	McAvoy and Butler (2009) Harris, Collins, and Hevner (2009a) ** Maruping, Viswanath, and Agarwal (2009) Adolph, Kruchten, and Hall (2012) Persson, Mathiassen, and Aaen (2012) Goh et al. (2013) Wagner et al. (2013) Gregory, Sambhara, and Mathiassen (2013) Santos et al. (2014)	Harris, Hevner, and Collins (2009b) Iivari and Iivari (2011)
ASD Use and outcomes	Bellini et al. (2005) Sfetsos et al. (2006) Arisholm et al. (2007) Choi et al. (2008) Pikkarainen et al. (2008) Yadav et al. (2009) Sfetsos et al. (2009) Balijepally et al. (2009) Vidgen and Wang (2009) Lee and Xia (2010) Hannay et al. (2010) Hong et al. (2011) Ramasubbu, Kemerer, and Hong (2012) Daneva, et al. (2013) Melo, et al. (2013) Wood et al. (2013) Hollis and Maiden (2013) Schmidt, Kude, Heinzl, and Mithas (2014) Hummel and Rosenkranz (2014)	Meso and Jain (2006) Trinidad et al. (2008) Port and Bui (2009) Batra, VanderMeer, and Dutta (2011) ** Alaa and Fitzgerald (2013) Schmidt, Kude, Tripp, Heinzl, and Spohrer (2013) Babb et al. (2014) Boschetti et al. (2014) Drechsler and Trepper (2014) Dissanayake, Dantu, and Nerur (2014)

(Table 2.2 continued)

Theme	Empirical	Conceptual
Human and Social Factors in ASD	Dingsøyr and Hanssen (2002) McAvoy and Butler (2007)** Hadar, Sherman, and Hazzan (2008) Sharp and Robinson (2008) Acuna et al. (2009) Maruping, Zhang, and Venkatesh (2009) Moe et al. (2010) Hoda, Noble, and Marshall (2011) Strode et al. (2012) Ryan and O'Connor (2013) Hoda, Noble, and Marshall (2013) McAvoy, Nagle, and Sammon (2013) Hansen and Lyytinen (2014) Bishop and Deokar (2014)	Holz and Maurer (2002)** Sena and Shan (2002) Doran (2004) Crawford et al. (2006) Balijepally and Nerur (2006) Layman et al. (2008) Salazar-Torres et al. (2008) Scheerer (2014) Yu and Petter (2014)
ASD Foundations and Non-specific themes	Mafakheri et al. (2008)* Johannessen and Ellingsen (2009)* Tiwana (2010)* Falessi et al. (2010) Baskerville, Pries-Heje, and Madsen (2011) Hanssen (2012)* Keith, Demirkan, and Goul (2013)*	Zimmer (2003)* Fang et al. (2004)* Soch and Walter (2006) Northover et al. (2006) Nerur and Balijepally (2007) Xu and Ramesh (2007)* Conboy (2009) Levardy and Browning (2009) Kakar (2014)

* = Agile software development is not the main focus
** = Multiple theoretical lenses

Agile Themes

In this section, a thematic analysis of the theoretical assessments found in our literature search was performed in order to answer RQ2—"*What insights have theory testing/theory building approaches provided on ASD methods?*" Of the 98 theoretical assessments that remained after Q2, the following themes were observed: (1) introduction and assimilation of ASD methods, (2) organizational factors and governance in ASD, (3) ASD use and outcomes, (4) human and social factors in ASD, and (5) ASD foundations

and non-specific themes. Table 2.2 displays the empirical and conceptual articles across each theme. Table 2.3 displays a breakdown of the empirical articles across each of the five themes.

Table 2.3. Agile Software Development Themes by Level of Analysis in Empirical Studies				
Theme	Org	Project	Team	Individual
Introduction and Assimilation of ASD Methods	Senapathi and Srinivasan (2012, 2014)	Cao, Peng, and Ramesh (2009) Mangalaraj, Mahapatra, Nerur (2009)	Conboy, Pikkarainen, and Wang (2007) Karlsson and Ågerfalk (2009)* Sarker and Sarker (2009) Wang, Pikkarainen, and Conboy (2012) Russo, Shams, and Fitzgerald (2013)	Drury, Conboy, and Power (2012) Overhage and Schlauderer (2012) Vijayasarathy and Turk (2012)
Organizational factors and Governance of ASD	Adolph, Kruchten, and Hall (2012) Gregory, Sambhara, and Mathiassen (2013) Wagner et al. (2013) Santos et al. (2014)	Harris, Collins, and Hevner (2009) Goh et al. (2013)	Maruping, Viswanath, and Agarwal (2009) McAvoy and Butler (2009) Persson, Mathiassen, and Aaen (2012)	

(Table 2.3 continued)

Theme	Org	Project	Team	Individual
ASD Use and outcomes		Daneva, et al. (2013) Hollis and Maiden (2013) Hummel and Rosenkranz (2014)	Bellini et al. (2005) ++ Sfetsos et al. (2006) ++ Arisholm et al. (2007) ++ Choi et al. (2008) ++ Pikkarainen et al. (2008) Balijepally et al. (2009) ++ Sfetsos et al. (2009) ++ Vidgen and Wang (2009) Hannay et al. (2010) ++ Lee and Xia (2010) Ramasubbu and Kemerer (2012) ++ Melo et al. (2013) Wood et al. (2013) Schmidt et al. (2014)	Yadav et al. (2009) Hong et al. (2011)
Human and Social Factors in ASD		Dingsøyr and Hanssen (2002) Hansen and Lyytinen (2014)	McAvoy and Butler (2007) Sharp and Robinson (2008) Maruping, Zhang, Venkatesh (2009) Acuna et al. (2009) Moe et al. (2010)	Hadar, Sherman, and Hazzan (2008) Bishop and Deokar (2014)
Theme	Org	Project	Team	Individual

(Table 2.3 continued)

			Hoda, Noble, and Marshall (2011) Strode, Huff, Hope, and Link (2012) Ryan and O'Connor (2013) McAvoy, Tadhg; Sammon, David (2013) Hoda, Noble, and Marshall (2013)	
ASD Foundations and Non-specific	Johannessen and Ellingsen (2009) Hanssen (2012)	Mafakheri et al. (2008) Tiwana (2010)	Baskerville, Pries-Heje, and Madsen (2011)** Keith, Demirkan, and Goul (2013)***	Falessi et al. (2010)
* = Method level ** = Multiple levels of analysis *** = Departmental level of analysis ++ = Pair programming teams				

Theme 1: Introduction and Assimilation of ASD Methods

The studies in this theme deal specifically with the introduction and assimilation of ASD methods. A common characteristic among these studies is their view of ASD methods as a form of software process innovation (Baskerville et al. 2003). All eleven empirical studies examine factors that influence ASD assimilation in organizations, seven of which make use of diffusion of innovation theory. A common theme among that emerges from these studies is that a variety of social and technical factors from across an organization may impact the adoption of ASD methods. For example, Chang and Thong (2009) conceptualized a framework of the impacts of ability-related, motivation-related, opportunity-related, and ASD methodology characteristics on knowledge management

outcomes, and their subsequent impacts on ASD use (acceptance). Mangalaraj et al.
(2009) developed and tested a model of antecedents --- individual, team, technological,
task, and environmental factors on ASD acceptance. This study found that such factors
influence the acceptance of XP practices in an organization. While Mangalaraj et al.
(2009) studied the assimilation of ASD practices across different XP practices within the
same company, Wang et al. (2012) studied the assimilation of ASD practices across
multiple organizations. Their study found that the length of use of ASD practices did not
proportionately affect assimilation depth and adopting teams do not always move through
the assimilation stages in a linear manner. Subsequently, Overhage and Schlauderer
(2012) focused on the long-term acceptance of ASD methods. Their study found that
developers perceived that Scrum required more discipline than traditional, waterfall
methods. In summary, the focus of studies listed above, and the models contained therein,
has largely been to further understand the antecedents to ASD usage. Only Senapathi and
Srinivasan (2012) measure the impacts of ASD usage on outcome variables with their
Agile Usage Model. Their original model has since been refined, expanded, and tested
(Russo, Shams, and Fitzgerald 2013; Senapathi, Drury, and Srinivasan 2013; Senapathi
and Srinivasan 2014). One way in which researchers can extend the Agile Usage Model
is to test additional ASD outcomes such as time to market and cost. Another is way to
extend the model is to test significant moderators such as internal versus external product
software development or organizational factors like culture or management structure.

In terms of levels of analysis, empirical studies are spread over the individual, team,
project, and organizational levels. At the individual level, two studies examine inhibitors

to the adoption of ASD methods, namely, ineffective decision making and negative perceptions from developers (Drury et al. 2012; Vijayasarathy 2012). Of the four studies at the team level, three examine post-adoptive ASD adoption while one explores ASD values in method configuration. At the project level, one study presents a framework for adapting ASD methods and another examines the acceptance of software process innovations (Cao et al. 2009; Conboy et al. 2007). Together these articles show that the adoption of ASD methods encompasses factors at multiple organizational levels, and not just the development team level. However, our search results did not find any multi-level studies on ASD adoption. Thus, this is an area for future research. One recommendation is that researchers may employ theoretical lenses like adaptive structuration theory (AST) (Cao et al. 2009) to highlight existing sources of organizational structure that impact the successful adoption of development processes. Because organizations contain unique corporate environments, firms must consider not just the internal processes the related to single ASD techniques. Rather, organizations need to understand how ASD practices must be tailored to fit within a specific organizational context without compromising ASD's fundamental characteristics. Theoretical lenses like AST may be useful for firms that are considering the transition from waterfall to ASD methods because it underlines existing corporate structures and challenges to ASD implementation.

Conceptual studies in this theme vary in terms of study objectives, for which we were unable to identify common subthemes amongst them. The six conceptual studies include: refinements to prior iterations of the Agile Usage Model; examinations of the differences in the selection of agile and traditional methods; and a framework for assessing both the

degree of agility required and how to identify appropriate ways to introduce this agility into an organization.

One area is need of expanded research that highlights the relationship between organizational factors and the adoption of ASD methods. For instance, some studies include cross case analyses of ASD teams across different corporate context and corporate structures. However, the impacts of those differing factors on adoption remain largely unaddressed. Specifically, differences in ASD diffusion between in-house development departments within organizations and software vendors remain unclear. In addition, identifying the factors effecting the adoption of ASD methods in GSDs is another area in need of expanded research. Finally, another area of expanded research is understanding of the assimilation gaps between non-adopters, adopters, and mature teams. While prior research has examined the ASD assimilation by exploring the adoption of individual ASD techniques across ASD teams (Conboy et al. 2007), further examination is needed to determine the social and technical factors that influence ASD assimilation decisions among teams, projects, and organizations across the assimilation stages.

Theme 2: Organizational factors and Governance

All eleven studies deal with the enactment of controls used to govern ASD teams, five of which employing control theory. Control theory, which finds its roots in engineering and mathematics, involves the manipulation of parameters that affect the ability of dynamic systems to produce a desired or optimal outcome (Weisstein 2012). When applied to the

study of ASD teams, control theory offers insight on ASD teams by highlighting the effects of enacting various control modes on behaviors during the development process. While some studies in this theme focus more on the manipulation of project controls and their impact of project outcomes under various conditions (Maruping et al. 2009a), others focus on a deep understanding of controls (Persson et al. 2012; Gregory, Sambhara, and Mathiassen 2013), freedom (Wagner et al. 2013), and the relationship between control and flexibility (Harris et al. 2009b) in the governance of ASD teams. For example, Maruping et al. (2009a) used control theory to identify the contingencies affecting the efficacy of different control modes (Agerfalk et al. 2009). Their findings suggest that autonomy is most effective when given to the team as whole, as opposed to individuals within the team; and it is given along with specific performance targets for the team (Maruping et al. 2009a). Harris et al. (2009a) conceptualized a framework for controls in flexible software development. Later, the authors operationalize their framework by combining control theory and dynamic capabilities theory to examine ASD teams under uncertain conditions (Harris et al. 2009b). Their findings suggest that the degree of flexibility required in a software project depends upon the conditions therein, but that flexibility may be needed when the starting conditions are uncertain. Taken together, these studies suggests that a portfolio of informal and formal controls is the most appropriate approach for governing ASD teams and unlocking team capabilities (Harris et al. 2009b; Goh et al. 2013).

At the organizational level, there were four empirical studies and one conceptual study. Three studies examined the relationship between organization factors such as

organizational strategy, organizational culture, and the use of ASD methods. First, Iivari and Iivari (2011), the sole conceptual study in this theme, used the competing values model to examine the relationship between organizational culture and the deployment of ASD methods. They propose a number of hypotheses to investigate the relationship between organizational culture and ASD methods deployment. Second, two studies examined how knowledge is managed and shared in ASD environments (Adolph, Krutchten, and Hall 2012; Santos et al. 2014). These studies suggest that there are issues that impact intra-team and inter-team communication as ASD teams operate in a business context. Third, the remaining two studies deconstruct previous notions of control and flexibility. Gregory, Sambhara, and Mathiassen (2013) extend the use of control theory in ASD research by developing chains of control in ASD to further represent how control is represented in software development. Using a deconstructed framework, their findings suggest that controls are revealed across the development process and across multiple representations of organizational levels of control. Overall, the studies listed above suggest that the control-flexible balance in ASD teams must be resolved in multiple organizational levels and the identification and impact of controls is widespread and not confined to the team level. However, Wagner et al. (2013) challenge the view that the freedom that developers experience when using ASD methods is a balance that is negotiated from structure. Instead, they suggest that freedom is made from the very structure that is often viewed as a constraint in traditional methods.

As previously noted, two studies at the project level examine the relationship between enacting controls and unlocking capabilities. At the team level, two articles focus on

enacting controls, one in GSD (Persson, Mathiassen, and Aaen 2012) and another under changing user requirements (Maruping, Venkatesh, and Agarwal 2009). In addition, McAvoy and Butler (2009) study the role of project management in ineffective decision making within ASD projects.

At the individual level, our analysis did not yield any studies, indicating a clear area for future research. Similar to prior research on the impact of antecedents on requirement analysis' success at the individual level (Yadav et al. 2009), the use of control theory can be leveraged to aid researchers in understanding the impacts of certain types of control mechanisms on the perspectives of individual developers. The examination of how formal and informal controls are implemented from an individual perspective may provide managers with insights about the attitudes and outlooks that developers bring to the software-building process.

Another area in need of further study is the role of management in ASD teams. Some ASD methods, which promote the notion of self-organizing teams also frown on micro-managing. However, little research has focused on how to macro-manage ASD methods, especially in agile transformations, or widespread ASD usage situations. In turn, the changing of job roles needs to be clarified as team's transition to ASD methods. For example, how does an organization manage traditional project management roles during ASD usage? Does a project management now become a Scrum Master? The answer to these questions is under rigorous debate. McAvoy and Butler (2009) suggest that the project managers can play the role of devil's advocate in order to facilitate effective

decision making in ASD teams. Future research should expand the role of middle management's relationship with ASD teams.

Conceptual studies in this theme include the relationship between organizational culture and the deployment of ASD methods (Iivari and Iivari 2011), and the use of controls in flexible software development projects (Harris et al. 2009b). The latter was developed into an empirical examination, while the former has not and remains an opportunity for future research. In fact, two practitioner surveys, Version One (2014) and Ambler (2014) cite organizational culture as the top challenge of ASD adoption, yet only one study in this theme investigates this relationship. Furthermore, we highlight the dearth of studies on organizational factors and the post-adoptive use of ASD methods. Researchers will need to consider the relationship between organizational level factors, such as company culture, and ASD assimilation to understand how these factors may affect ASD usage.

Theme 3: ASD Use and Outcomes

In this theme, a total of 29 studies were categorized as focusing on ASD use and outcomes. Studies in this theme focus on either ASD usage and its impact on project outcomes or the internal processes at work during ASD use. The 19 empirical studies were subdivided by those studies that examine single ASD techniques (i.e. pair programming and requirements gathering) and those that examine ASD usage more holistically and its impact on project outcomes. A common theme that can be observed from these studies is their view of ASD as a process, as described in the Input-Process-Output (I-P-O) model, as opposed to a process innovation as discussed in theme 1. In

terms of single techniques, eight empirical studies employ the use of theoretical perspectives to examine the practice of paired programming. All of these studies measure the impacts of pair programming on outcome variables. Moreover, two empirical studies examine requirements prioritization, another ASD technique, in globally distributed contexts.

Conversely, eight empirical studies examine the impact of ASD methods on outcomes, in a general sense, as follows: three on team performance, one on team productivity, one on information systems development (ISD) success, one on communication, one on user acceptance of ASD, and one on creativity. Of these, five studies use the input-process-output model for teamwork effectiveness to study traditional team outcomes of performance and productivity. For example, both Melo et al. (2013) and Wood et al. (2013) use I-P-O to study the impact of post-adoptive use of ASD methods on team outcomes. Melo et al. (2013) confirmed the findings of prior research (Parolia et al. 2007) by suggesting that coordination processing affects the establishment of common goals, which, in turn, impact team performance. Interestingly, their study also suggested that rigid organizational structures increased the negative impact of inter-team coordination processes on team productivity. Wood et al. (2013) study suggest that influences in client and team focus in XP software projects were a result of using the methodology and not merely an enhanced use of teamwork. Hummel and Rosenkranz (2014) use I-P-O to explain the flow and impact of communication on ISD success, which is operationalized as process performance, and user satisfaction. Although the Lee and Xia (2010) do not specifically mention the I-P-O model by name, their model of ASD team effectiveness

follows the I-P-O framework. Overall, the I-P-O model provides a simple framework for guiding research aimed at testing the impact of inputs of ASD use on outcomes. Their findings suggest that software team response extensiveness has a positive effect only on scope whereas software team response efficiency has a positive effect on team performance as measured by time, budget, and scope. Three other empirical studies focus on the internal flow of decision making and communication (Pikkarainen et al. 2008), the user acceptance of ASD (Hong et al. 2011), and the extension of the epic process on creativity in requirements (Hollis and Maiden 2013).

Finally, Vidgen and Wang's (2009) empirical study uses complex adaptive theory (CAS) to study the internal team processes that enable and inhibit agility. Their findings recognize the ability of ASD teams to collaborate with customers to coevolve business value, work in a rhythmic pace, learn collectively, adapt development processes, and to create product innovations, in comparison to traditional methods. Although CAS is among the most applied theories for studying ASD (Meso and Jain 2006; Levardy and Browning 2009; Batra, VanderMeer, and Dutta 2011; Alaa and Fitzgerald 2013), Vidgen and Wang's (2009) study is the only empirical examination of ASD using CAS found in this study. In addition, CAS is the only theoretical perspective in this study that theorizes self-organizing ASD teams from holistic system perspective. The application of CAS at the team level offers fruitful insight into the inner workings of the ASD development process as autonomous agents work in unison with the ability to alter their course of action by effectively sensing their environment. Further empirical examination is merited in order to expand the relationship between CAS and ASD. Future research might answer

the question: In what sense or to what extent, is ASD a complex adaptive system? Considering the likely overlap between ASD and CAS, future research could highlight commonalities and differences between the two.

No organizational level studies were found in this theme. Although the ASD methods originated from the software development practice as a team level concept, its widespread usage in large organizations is understudied. Further research can focus on the impact of a series on ASD teams at the organizational level as a part of an organizational strategy. Moreover, the link between such a strategy and enterprise agility needs further study. With the expanding use of ASD methods, it will become increasingly important to understand the impacts of organization wide ASD use.

Two studies at the project level and six studies at the team level all measure ASD use on project or team outcomes. Two studies at the individual level, examined the individual perceptions of requirements and the individual user acceptance criteria of ASD projects respectively. However, no theoretical perspectives focus on individual developer perceptions of the use of ASD methods on project outcomes.

The ten conceptual studies in this theme vary widely in terms of theoretical perspectives and study focus as shown in Table 2.2. Only complex adaptive theory, used three times, appeared in multiple studies (Meso and Jain 2006; Batra, VanderMeer, and Dutta 2011; Alaa and Fitzgerald 2013). Four method level studies focused on the efficacy of the ASD process (Trinidad et al. 2008; D. B. Port, Tung 2009; Boschetti et al. 2014; Babb 2014).

Overall, studies in this theme contained the most conceptual theoretical assessments, which suggest that there may be room for further empirical examination of ASD. Only two of the twelve theoretical perspectives in these conceptual studies were employed in empirical investigations elsewhere.

Theme 4: Human and Social Factors in ASD

A total of 22 studies containing 24 theoretical perspectives examined human and social factors involved in ASD. The 14 empirical studies were categorized under the following sub-themes: cognition (7), coordination (5), and personality (2). A common theme among these studies is their emphasis on sociocultural factors that influence or explain ASD processes. Under the cognition subtheme, each study focuses on the process of acquiring knowledge and understanding during ASD. First, Dingsøyr and Hanssen (2002) found that the use of post-mortem techniques can be used to adapt XP. Second, McAvoy and Butler (2007) examined the Abilene Paradox on double-loop and triple-loop learning in ASD teams. Their findings suggest that learning is more than the cognitive process of acquiring a new skill, but also involves changes in behavior and beliefs. Third, Sharp and Robinson (2008) employed the theory of distributed cognition to study two ASD artifacts: story boards and the Wall during ASD. They conclude that story boarding is significant in the underpinning the highly collaborative and self-organizing style of ASD teams. Fourth, McAvoy et al. (2013) use the theoretical lens of mindfulness to examine ISD agility. They argue for the importance of mindfulness as a prerequisite for ISD agility that can be used to identify ISD team members. Their findings suggest prerequisites for ISD agility can be identified through examining the behaviors of the

software team members and not just the software practices. Fifth, Ryan and O'Connor (2013) study the acquisition and sharing of tacit knowledge in ASD. The study's findings suggest that team tacit knowledge is acquired and shared directly through high quality social interactions. Furthermore, both transactive memory and team tacit knowledge were found to predict effectiveness but not efficiency in software teams. Finally, Hansen and Lyytinen (2014) analyze requirements cognition in multiple development paradigms, including ASD, using the theory of distributed cognition. The findings suggest that distributed cognition principles offer fruitful insight in evaluating how the change in distribution affects requirements activities and their outcomes. All of the six cognitive studies focus on the team or individual project level. In a broad sense, these studies highlight the cognitive mechanisms that facilitate the management and transfer of knowledge in ASD.

Under the coordination subtheme, expertise coordination was used to study the role of collective ownership and coding standards for developers in ASD teams (Maruping 2009b). They show the positive role of the two practices in moderating the relationship between expertise coordination and software project technical quality, with collective ownership attenuating the relationship and coding standards strengthening the relationship. More broadly, first, Strode et al. (2012) used coordination theory to study the coordination of ASD practices in co-located teams. They show how an ASD project coordination strategy is achieved to improve coordination effectiveness. Second, Moe et al. (2010) develop a teamwork model to understanding ASD teams. Their model suggests that transitioning from individual work to self-managing teams requires a reorientation

not only by developers but also by management. We leverage this finding to highlight the organizational impact of ASD methods in the development of both summary framework at the end of this section and our theoretical model in section 5. Third, Hoda, Noble, and Marshall (2011, 2013) develop theoretical frameworks concerning self-organizing teams using grounded theory. They identify important roles that make ASD teams self-organizing. As a whole, these studies highlight the roles, responsibilities, and alignment of the people and processes involved in ASD.

Studies under the personality subtheme focus on the individual preferences for ASD (Bishop and Deokar 2014) and the impact of team member personalities on job satisfaction and quality (Acuna et al. 2009). These studies emphasize the role and impact of inputs at the individual level. Their findings suggest that personality may be a key factor in achieving ISD team diversity.

As shown in Table 2.3, no studies were found at the organizational level and only one study was found at the individual level. Thus, there is opportunity for further study at each level. We recommend that organizational level constructs including organization learning and organization memory may yield helpful key insights in the way that collective entities embrace, resists, and implement ASD methods. As large companies undergo agile transformations, organizational factors may unlock key insights pertaining to the adoption of an agile mindset within a given context. Additionally, empirical theoretical assessments at the individual level can yield insights into the mindsets of individual developers during such a transformation.

Our search yielded 10 conceptual theoretical assessments, six of which use knowledge management to study ASD. These knowledge management studies are found primarily in the book Advances in Learning Software Organizations, Proceedings (2002, 2004). Others conceptual studies focus on the personality of team members on productivity (Balijepally and Nerur 2006; Layman et al. 2008), the shared understanding between developers and customers (Yu and Petter 2014), and coordination in large ASD teams (Scheerer 2014).

Thus, the same three themes emerge in the conceptual studies as in the empirical studies. This suggests that the human interactions highlighted by the coordination of team members, their ability create, retain, and transfer knowledge, and their individual personalities play an important role in ASD processes.

Theme 5: ASD Foundations and Non-specific

A total of 16 ASD studies did not relate strongly to any of our four themes. Thus, these studies were classified as non-specific as topics in this theme varied with few commonalities between studies. Topics that were discussed in multiple studies include 1) underlying theoretical assumptions of ASD (Northover et al. 2006; Nerur and Balijepally 2007), and 2) defining agility (Conboy 2009; Baskerville, Pries-Heje, and Madsen 2011). Despite their small number, these studies raise important issues concerning the nature and theoretical foundation of ASD methods. Northover et al. (2006) contrasts the ideas of Thomas Kuhn against the ideas of Karl Popper in relationship to the shift from traditional to ASD methods. The authors point out that Popper's concepts of falsification and error

elimination have a stronger affinity with core ASD concepts, such as the iterative test-units found in Extreme Programming (XP) than Kuhn's view of paradigm shifts. Thus, they conclude that ASD methods are more related to positivist's notions of fact checking rather than being a different epistemology of pragmatism as later suggested by Nerur and Balijepally's (2007). The latter argues that ASD represents new epistemology of software development because of its people-centered values and may in fact be "a theoretical".

Concerning the definition of agile, Conboy (2009) proposed a definition based off a literature review of fields outside of software development. Later, Baskerville, Pries-Heje, and Madsen (2011) examined the evolution in the meaning of "agile ISD" over time. These studies seek to understand what constitutes ASD. They also show the need for further research to understand the theoretical foundations of ASD.

The remaining twelve studies consist of various unrelated topics such as software process tailoring, graph theoretical indicators and refactoring, and emerging software ecosystems. In addition, the number of empirical to conceptual studies was evenly split at eight each. The next section synthesizes the themes above into a table (see Table 2.4) and summarizes key findings.

Summary

Table 2.4. Themes in ASD Research					
Level of Analysis	Theme 1: Introduction and Assimilation of Agile Methods	Theme 2: Organizational factors and Governance of ASD	Theme 3: ASD Use and outcomes	Theme 4: Human and Social Factors in ASD	Theme 5: Agile Foundations and Non-specific
Organizational	☑	☑	☒	☒	n/a
Project	☑	☑	☑	☑	n/a
Team	☑	☑	☑	☑	n/a
Individual	☑	☒	☑	☑	n/a

Based on a thematic analysis of 98 theoretical assessments, 90 articles identify at least one theoretical perspective used to study ASD methods. Eight articles were removed because ASD was not the main focus. Table 2.4 to helps us answer RQ3, "Where are the gaps in our theoretical understanding of ASD research?" by clearly indicating the gaps in the extant literature in terms of the levels of analyses that are referenced under each theme. The ☒ indicates that no studies under a specific theme employ the corresponding level of analysis. There has been very little attention given to ASD usage or social factors at the organizational level nor the use of governance controls at the individual level as shown in Table 2.4. These gaps may signal the opportunity for future research, a mismatch between the theme and the usual ways in which topics are studied within the theme, and/or difficulties in executing studies at a given level of analysis such as a lack of data points at the organizational level. Nevertheless, we observe that the majority of early ASD studies focus on understanding the phenomenon by observing, interviewing, and surveying agile software developers from a team-level perspective. However, the

presence of studies at every level of analysis points to the widespread impact of the ASD phenomenon in an organizational setting. Additionally, research at the organizational level provides a broad view of ASD by emphasizing its function as an interconnected piece of the organizational puzzle. However, data collected at the organizational level does not explain the inner workings of the ASD processes in as much detail as lower levels of analysis do. The individual level complements higher levels of analysis in explaining how a single developer's perspective affects ASD processes in organizations. Taken together, theoretical assessments at all levels have added significant contributions to the body of knowledge.

Outside of the clear findings in Table 2.4, we summarize what is known and what is not known as shown in Table 2.5. Overall, the relationship between the adoption and use of ASD methods and its outcomes within a given corporate context, including how social groups interact with ASD implementations appears to be of strong interest to both researchers and practitioners.

Table 2.5. Summary of what is known versus what is not known
Theme 1: Introduction and Assimilation of Agile Methods
The extant literature suggests that length of ASD use does not proportionately effect assimilation depth (Senapathi and Srinivasan 2012). Therefore, ASD use is conceptualized as both the extent and intensity of use in organizations. In terms of antecedents to ASD use, the relative advantage of ASD methods over its predecessor is among the most influential factors that affect ASD use (Senapathi and Srinivasan 2012). However, the validation of other predictors of ASD use that are empirically supported across multiple studies is lacking.
In addition, our understanding of which antecedents impact ASD use during specific assimilation stages and across contexts is limited. Knowledge is especially limited at the routinization and infusion stages. The lack of validated antecedents is further emphasized when considering that the impact of some antecedents may differ during specific assimilation stages and across contexts. Little research has been conducted to

(Table 2.5 continued)

investigate the changing impact of antecedents at different stages of the assimilation process. Thus, there is little understanding of the constituents that affect the impact of the antecedents to ASD use.

In terms of outcomes of ASD use, predictability, productivity, quality, and customer satisfaction have been validated in prior studies (Coa et al. 2009; Senapathi and Srinivasan 2014). However, additional impacts of ASD use found in practitioner literature such as time to market have not been tested.

Theme 2: Organizational factors and Governance

The extant literature highlight the presence of several control types found throughout the ASD development process. However, most studies map controls to the XP methodology (Maruping et al. 2009a; Harris et al. 2009). Further research is needed to broaden understanding of the role of controls across additional methods and to understand the dynamics of controls across different phases of IS projects. Moreover, prior studies suggest that ASD use leads to higher software quality under changing requirements and specific control modes. More research is needed to understand the contingencies under which the uses of specific control types influence project outcomes.

The literature suggests that the use of ASD methods enables flexibility in the software development process, which is needed when the starting conditions are uncertain. Knowledge is limited concerning additional ways of enhancing software development team flexibility.

Theme 3: ASD Use and Outcomes
As previously mentioned in theme 1, a number of inputs (e.g. team autonomy, team diversity, etc.) that impact ASD use, which, in turn, impact a number of team and project level outcomes (e.g. performance, productivity, etc.) have been identity in prior research. Ironically, there is a dearth of validated surveys on ASD use. Additionally, the impact of widespread ASD use on the organizational level constructs (e.g. organizational agility) is limited.

Theme 4: Human and Social Factors in ASD
The extant literature suggests that ASD techniques and artifacts promote various aspects of teamwork. Additionally, communication, cognition, and coordination are the active ingredients of ASD use. Unfortunately, our understanding of the interaction between organizational level factors and social factors is limited.

Theme 5: Agile Foundations and Non-specific
There is some debate about the philosophical assumptions of ASD methods. Some studies classify ASD methods as a positivist approach to software development, while others classify ASD as more pragmatic. Further research is needed in order to understand how these differences might affect our conceptualizations of ASD methods. Additionally, the definition of ASD and the meaning of the word "agility" have evolved over time.

In relation to this interest, we discuss the need for more understanding of post-adoptive ASD use, the different perspectives on ASD use, the role of influencers to ASD use, and the implications of these elements on future research. In the next section, we highlight the contributions and limitations of each theme taken separately before introducing a model that integrates key concepts within each theme and motivates future research based on the results from the thematic analysis.

Post-adoptive Agile Use

Post-adoptive agile use refers to the stages in which an innovation is being used. In the innovation literature on ASD methods, the acceptance, routinization, and infusion stages are considered to be post-adoptive as shown in Figure 2.4.
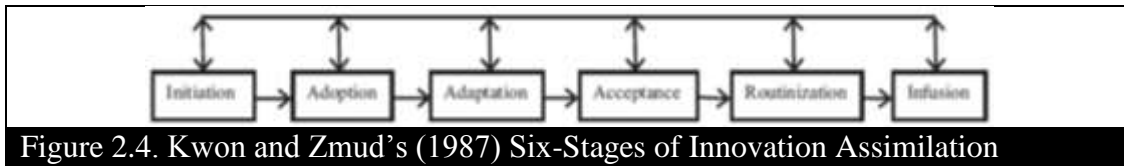
Figure 2.4. Kwon and Zmud's (1987) Six-Stages of Innovation Assimilation

Although the extant literature underscores the reasons why ASD methods are initially adopted (Tessem 2003; Bahli and Zeid 2005; Svensson and Höst 2005), there is a dearth of empirical investigation concerning the use of ASD beyond the initial stages of use despite calls for a greater understanding of post-adoptive issues (Abrahamsson et al. 2009). Regardless of the widespread adoption of ASD methods across the software development industry, many organizations struggle with maintaining ASD methods in the long-term. They are challenged to align their people, processes, and tools to those of ASD methods. Further research of the downstream phases of ASD use is needed to understand how to sustain ASD use in the long-term. To fill this gap in our knowledge,

we first highlight the two dominant perspectives of ASD use in the literature in the next section.

Perspectives on Agile Use

The extant literature conceptualizes ASD use either as the assimilation of a software process innovation using the diffusion of innovation theory (theme 1) or as a software development method that has an effect on project outcomes using the I-P-O model (theme 3). We discuss each perspective below.

Agile Use (I-P-O): Theme 3 conceptualizes ASD use as one of either being used or not used. Additionally, the application of the I-P-O model in ASD research do not account for the organizational level as shown in Table 2.4. Although most measures of ASD use account for the extent to which certain ASD techniques (pair programming, refactoring, etc.) are implemented within a given ASD practice (Scrum, XP, etc.), these measures do not account for the extent to which ASD practices are used throughout the organization. For instance, Schmidt et al. (2014) examines the effect of ASD use on markers of adaptive team performance. ASD use is measured by the extent to which a team implements code reviews and other techniques of the Extreme programming practice. We compare this notion of ASD use as shown in Figure 2.5 with the assimilation perspective described in the next section.
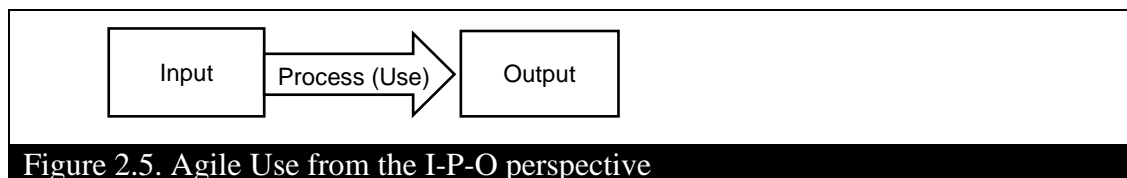
Figure 2.5. Agile Use from the I-P-O perspective

Agile Use (Assimilation): Theme 1 conceptualizes ASD use from an assimilation perspective, which takes the intensity and extent of use into account. Figure 2.4 displays the six-stage model of innovation assimilation (IA) (Kwon and Zmud 1987), which describes the intensity and extent of use in across multiple stages (acceptance, routinization, and infusion). Moreover, the IA research extends the diffusion of innovation theory to study the diffusion of an innovation from an organizational standpoint. As a result, the assimilation perspective helps to fill the gap in theme 3, which neglects on organizational perspective. In addition to aiding in our understanding of the antecedents and consequences of ASD use as found in theme 3, studies in theme 1 elaborate on the organizational and human factors that impact the extent of ASD use in an organization. As shown in Table 2.4, theme 1 is the only theme that contains empirical studies at all four levels of analysis.

Summary: The combination of the assimilation and I-P-O perspectives of ASD use highlights the importance of understanding both the way that the use of ASD impacts project performance and the appropriation of these methods within the existing organizational environment. Additionally, we note that the assimilation perspective offers a useful framework for understanding post-adoptive issues, which are of benefit to practitioners, yet lacking in academic research. Together these perspectives point to the technical and social interactions that are involved during ASD, which is reminiscent of socio-technical research (Mumford 1983). As ASD methods encourage collaboration and a horizontal approach to completing tasks, ASD may face serious challenges in hierarchical, top-down organization structures and cultures. Moreover, the cooperation of

key stakeholders including customers plays a heavier role in the execution of agile versus traditional software development methods. We discuss such influencers of ASD use in the next section.

Influencers to ASD Use

Themes 2 and 4 focus on the organizational and social factors that influence the software development process in an organization setting. Theme 4 focuses on internal influences to ASD teams and theme 2 focuses on external influences to ASD teams. From a variance model perspective, these themes represent antecedents to ASD use, and mediating and moderating effects of ASD use on ASD outcomes. For example, Maruping et al. (2009), under the theme 2, found that different types of controls produced different outcomes in ASD projects. The study of these controls, informal and formal, is not unique to ASD, but they offer helpful insights regarding the governance of ASD teams. However, past researchers study the governance of ASD teams in a top-down manner that encompasses the organizational, project, and team levels, but neglect the individual level.

Similarly, theme 4 focuses on the human and social factors that affect team members during the software development process. Like most software development research, theme 4 focuses on issues that impact those directly responsible for delivering software such as coders, testers, and project managers. However, it largely ignores the role of the more senior level decision makers and the existing organizational structure that impact the development process.

Integrated View of ASD Use

Using the results of our thematic analysis, we seek to motivate ASD research moving forward. The results of our thematic analysis suggest that a more integrated view can be applied to the study of ASD methods to reduce fragmentation in our theoretical understanding. Figure 2.6 offers an integrated view using the major themes found in our study.
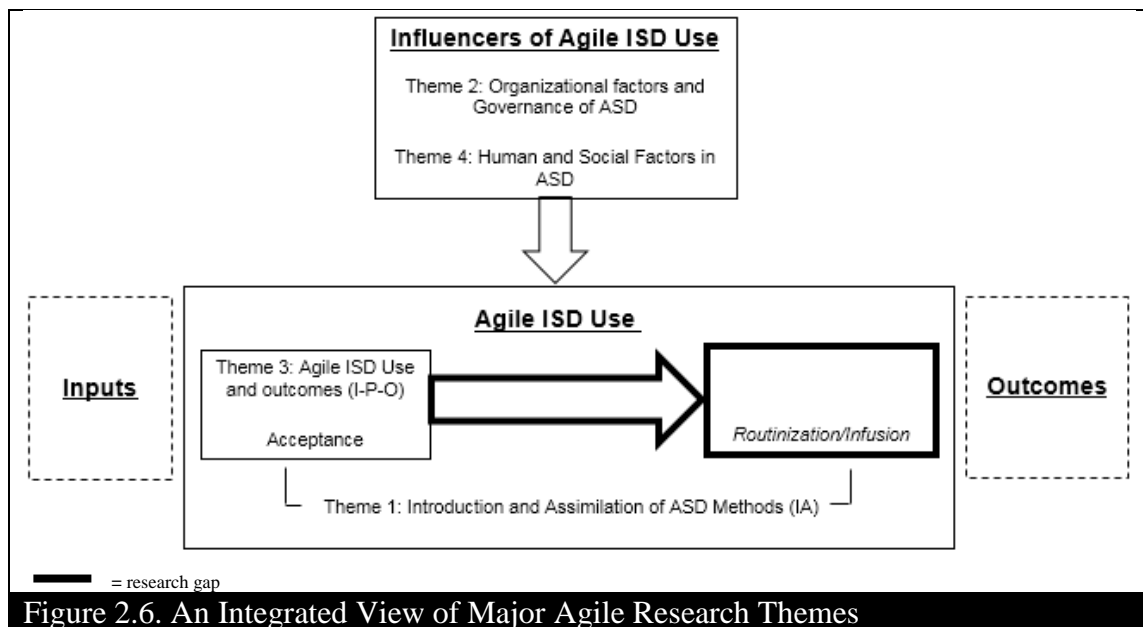
Figure 2.6. An Integrated View of Major Agile Research Themes

First, as noted in the last section, themes 2 and 4 can be understood as influencers to ASD use. Although the influences to ASD use described in these themes are studied separately in most of the extant literature, recent studies examine the interaction of these influences on ASD use (Santos et al. 2014). This view seems to agree with the call for a better understanding of ASD methods beyond the stages of initial use in the extant literature:

> Specific needs of organisations and human nature inevitably lead to diverse interpretations and implementations of a method, which in turn lead to different, sometimes surprising, effects and consequences of use of agile methods and associated practices. (Abrahamsson et al. 2009)

The particular needs of organizations and human nature are especially emphasized in software development because, unlike other forms of design such as constructing a building, software design lacks well-accepted, well-understood, and well-defined blueprints for programming (as a 50-60 year old practice) (Socha and Walter 2007). Instead, software can be built in multiple ways, with different methods, in an environment that is not subject to well understood physical laws. Therefore, the ways in which a group uniquely learns, communicates, and works together to solve problems would be of assistance in understanding how software teams perform the knowledge work of developing software.

Second, the assimilation perspective in theme 1 offers several advantages over the I-P-O perspective in theme 3 for studying ASD use. First, the assimilation perspective is better positioned to address the call for a better understanding of ASD methods beyond the stages of initial use in the extant literature. The assimilation perspective finds its roots in the diffusion of innovation theory and uses stage models that examine both the extent and intensity of use, which respects the nature of ASD use in practice. For instance, with respect to Figure 2.4 and Appendix A8, we can observe that empirical research is particularly lacking beyond the acceptance stage (routinization and infusion). This means observation in combination with findings in the practitioner literature (Sutherland 2014) suggests that there is a growing need to understand the factors that influence the long-term use of ASD methods as organizations implement ASD methods. The current body of knowledge calls for an understanding of post-adoptive use as follows:

> In contrast, the studies of issues associated with post-adoption use of agile methods are much less in number, even though there is increasing need to have a

better understanding of agile methods in use as many organisations have completed adoption stage and agile methods start to become well-established processes of these organisations. (Abrahamsson et al. 2009)

Second, results of our thematic analysis suggest that there is a need for a deeper understanding of the organizational constituents that affect the application of ASD in real-life contexts. Our thematic analysis shows that the majority of theoretical perspectives on ASD focus on ASD use and its resulting outcomes. However, prior IT research suggests that outcomes are not simply the product of use, but behavioral and organizational factors (Markus and Robey 1983). Additionally, only the assimilation perspective in theme 1 contains theoretical insights at all levels of analysis (see Figure 2.4). Therefore, we argue that the assimilation perspective offers a more context rich understanding of the phenomenon, which explicates ASD use by providing an understanding of the proliferation of ASD methods in an organizational setting.

Finally, the contributions of the I-P-O perspective are not irreconcilable with the assimilation perspective. In fact, the inputs and outcomes of ASD use can be integrated and modeled as found in the Agile Usage Model (Senapathi and Srinivasan 2012; 2014), which integrates inputs and outputs while measuring both the intensity and extent of use. Overall, the assimilation perspective provides a broader theoretical base to understand the effects of ASD methods in an organizational setting than the I-P-O perspective.

Given these key points, we developed the Theoretical Perspectives on Agile Software Development Framework (TPA) framework to provide further insight into the research

gaps from an assimilation perspective in reference to RQ3 (see Appendix A8). Drawing on the TPA and our analysis of the extant literature, we identify that organizational factors play a pivotal role in the diffusion of an innovation within an organization (Chan and Thong 2009), yet the understanding of their impact on the assimilation of ASD methods is limited. The investigation of this gap serves as the foundation on which we built a new model on to provide direction for future research.

The remainder of this research focuses on the relationship between organizational factors, particularly culture and structure, and the post-adoptive use of ASD methods. Not only are both of these issues significant gaps in the literature, or the "what", but we proposed that the intersection of organizational factors and post-adoptive ASD use helps explain "how" ASD is assimilated in an organizational setting and "why" certain conditions impede ASD assimilation. Although there are studies that highlight the impact of organizational factors on post-adoptive use, researchers have yet to explain the gaps between both non-use and initial use and most important to the current ASD practice, the gap between initial use and routinization. We now present a new model to guide future research on ASD routinization.

THEORIZING AGILE SOFTWARE DEVELOPMENT ROUTINIZATION

As previously discussed, the software development practice faces unique challenges that differentiate it from traditional forms of design. Challenges such as software team environmental uncertainty, changing user requirements, and the subjectivity in

interpreting user requirements increase the complexity of the software building process. ASD methods challenge the assumption that software requirements can be fully defined up front using a traditional, sequential approach. By taking an incremental approach, ASD teams relax the assumptions of traditional forms of design that rely on upfront planning, and instead, focus on understanding requirements as the artifact is being built in order to deal with change. Despite ASD benefits, routinizing ASD methods is difficult because of the existing assumptions that underlie the way group members relate to one another. From a design perspective, ASD adopts an emergent design approach (Beck et al. 2001), which does not conform to ways in which members communicate in the command and control cultures and structures found in many organizations. Thus, the conflict between ASD methods and the governing social structures and norms within an organization needs to be resolved in order to routinize ASD methods.

Theoretical model

Routinization is the stage of organizational assimilation in which an innovation is used as a normal activity (Kwon and Zmud 1987). At the routinization stage, ASD methods become an integral part of the software development process in an organizational setting (Cooper and Zmud 1990) and therefore, are no longer considered out of the ordinary (Wang et al. 2012).

The extant literature describes the adoption of individual technologies using Roger's (1962) diffusion of innovation theory (DOI), and then extends the application of DOI to the organizational-level (Rogers 1983; Kwon and Zmud 1987, Meyers and Goes 1988,

Cooper and Zmud 1990). We refer to the research on organizational-level assimilation of innovations as *assimilation theory* (Wang et al 2012). Drawing on assimilation theory, the agile literature describes ASD assimilation process using Kwon and Zmud's (1987) six-stage process of innovation assimilation, which begins with initiation and ends with infusion (Gallivan 2001; Wang et al. 2012) (see Figure 2.4). Thus, *ASD assimilation* can be defined as the extent to which the use of ASD methods diffuses across the organizational projects or work processes and becomes routinized in the activities of those projects and processes (Cooper and Zmud 1990; Wang et al. 2012). The ASD assimilation process, through which ASD methods are introduced, accepted, and become a company's philosophy of software development, often overturning an existing methodology, is progressive, involving buy in from many pertinent stakeholders. Prior research has used assimilation theory to create and test a model of ASD usage, define individual assimilation stages, and identify the extent that specific ASD techniques are used by ASD adopters (Senapathi and Srinivasan 2012; Wang et al. 2012). However, little research has been conducted to expose the gaps in the ASD assimilation process (Fichman and Kemerer 1999), and understand why some organizations elect to adopt ASD methods, not only initially (acceptance), but also on a continual basis (routinization). With respect to the gaps between the acceptance and routinization stages in the ASD assimilation process, there is a need for examination of the differences between those organizations that elect to routinize ASD methods from those that elect not to routinize after successful completion of the acceptance stage. In our study, we refer to this assimilation gap as the *ASD acceptance-routinization gap*. *Assimilation gaps* can be defined as the difference between the patterns of cumulative assimilation events of an

innovation across a population of adopters (Fichman and Kemerer 1999). Drawing again on assimilation theory, we revisit the notion of ASD adoption and suggest that ASD researchers must consider several ASD assimilation gaps that can be observed from the study of various levels of adopters and non-adopters. Interestingly, the extant literature on process innovations suggests that organizational culture plays a key role in the continual use of an innovation (Dubé 1998; Dubé and Robey 1999). In addition, the extant literature suggests the use of theoretical lenses to highlight existing sources of organizational structure that impact the successful adoption of development processes (Cao et al. 2009). In turn, we build towards a theory of ASD routinization by explaining how the influences of organizational factors (culture and structure) significantly impact the routinization of ASD methods.

Using Kwon and Zmud's (1987) six-staged process, the ASD literature relates adoptive use of an innovation to the initial three stages (initiation, adoption, adaptation) and post-adoptive use and implementation to the last three stages (acceptance, routinization, infusion). In this study, we will build a model that investigates the *ASD acceptance-routinization gap*, where *ASD acceptance* refers to the introductory adoption and employment of ASD methods as a process innovation for one or more software development projects (Vijayasarathy and Turk 2012; Wang et al. 2012), and *ASD routinization* refers to the use of ASD methods as a normal activity in an organization (Senapathi and Srinivasan 2012).

Our model, as depicted in Figure 2.7, describes a major assimilation gap between ASD acceptance and ASD routinization, that being the role of organizational factors, culture and structure.
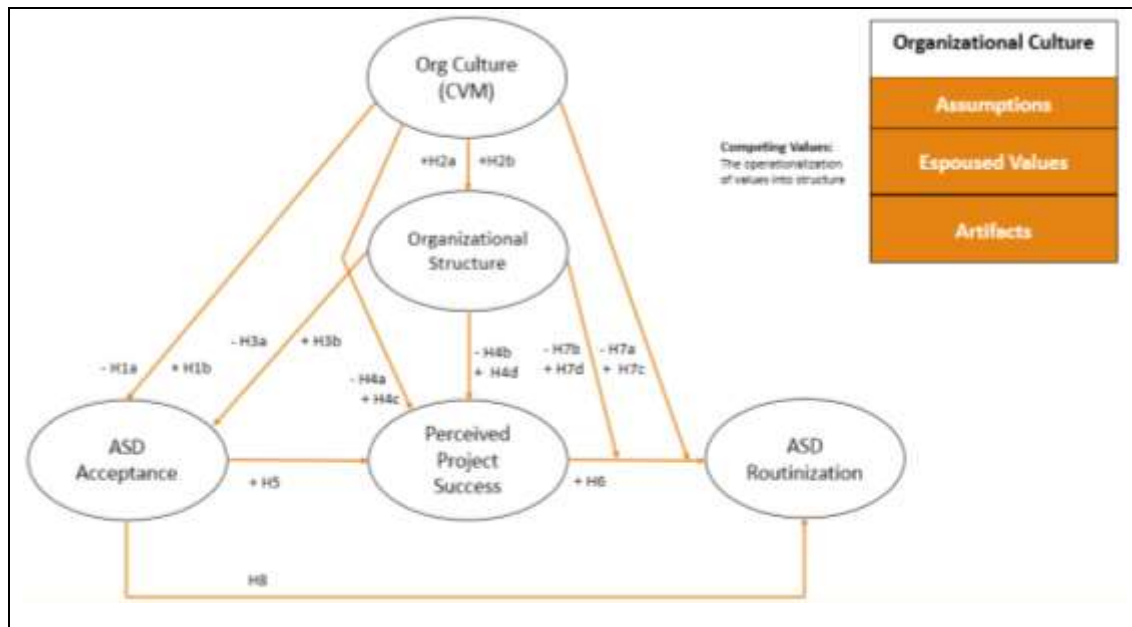
Figure 2.7. ASD Routinization Model

In the next section, we describe the constructs in our model --- *organizational culture, organizational structure, ASD acceptance, perceived ASD success, and ASD routinization* as shown in Table 2.6, and then build propositions (see Table 2.7) concerning the relationships therein to aid in our understanding of post-adoptive ASD use, and provide guidance for future research.

Table 2.6. Construct Definitions	
Organizational Culture	A pattern of shared basic assumptions learned by an [organization] as it solves its problems of external adaptation and internal integration, which has worked well enough to be considered valid and, therefore, to new members as the correct way to perceive, think, and feel in relation to those problems (Schein 1985).
Hierarchical Organizational Culture	A type of organizational culture that represents a form of organizing based on rules, bureaucracy, and formalization.

Development Organizational Culture	A type of organizational culture that represents a form of organizing based on innovativeness and adaptation.
Organizational Structure	"The logically consistent clustering of an organization's elements as it searches for harmony in its internal processes and consonance with its environment" (Mintzberg 1980).
ASD Acceptance	The introductory adoption and employment of ASD methods as a process innovation for one or more software development projects (Vijayasarathy and Turk 2012; Wang et al. 2012).
Perceived ASD Success	The extent to which an organization's ASD project(s) meets technical goals, remains within the budget, is delivered in time, and is accepted by the end user (Jiang, Klein, and Pick 2003; Procaccino and Verner 2006).
ASD Routinization	The usage of ASD methods as a normal activity in the organization; the innovation is no longer considered out of the ordinary (Wang et al. 2012).
ASD Assimilation	The extent to which the use of ASD methods diffuses across the organizational projects or work processes and becomes routinized in the activities of those projects and processes (Cooper and Zmud 1990; Wang et al. 2012).
Assimilation gaps	The difference between the patterns of cumulative assimilation events of an innovation across a population of adopters (Fichman and Kemerer 1999).

Table 2.7. Propositions and Hypotheses	
Proposition 1: A firm's organization culture, represented by its core values, will influence its acceptance of ASD methods.	H1a: A hierarchical organizational culture built upon values of stability and internal focus (bureaucracy) will tend to hinder the use of ASD methods. H1b: A developmental organizational culture built upon the values of flexibility and external focus (adhocracy) will tend to facilitate the use of ASD methods.
Proposition 2: A firm's organizational culture, represented by its core values, will influence its organization structure (form).	H2a: A hierarchical organizational culture will give rise to a functional organizational structure H2b: A developmental organizational culture will give rise to a projectized organizational structure.
Proposition 3: A firm's organizational structure will influence its acceptance of ASD methods.	H3a: Functional, bureaucratic organizational structures will tend to hinder the use of ASD methods H3b: Projectized organizational structures will tend to facilitate the use of ASD methods.
Proposition 4: Organizational	Negative Influence

(Table 2.7 continued)

factors, culture and structure, will influence perceived ASD success.	H4a: A hierarchical organization culture will negatively influence perceived ASD success. H4b: A functional organization structure will negatively influence perceived ASD success. Positive Influence H4c: A developmental organization culture will positively influence perceived ASD success. H4d: A projectized organization structure will positively influence perceived ASD success.
Proposition 5: A firm's ASD acceptance will influence its perceived ASD success.	H5: ASD acceptance will positively influence perceived ASD success.
Proposition 6: A firm's perceived ASD success will influence ASD routinization.	H6: Perceived ASD project success will positively influence ASD routinization.
Proposition 7: Organizational factors, culture and structure, will influence ASD routinization beyond the perceived success of ASD acceptance.	Negative Influence H7a: A hierarchical organization culture will negatively moderate the influence of perceived ASD success on ASD routinization. H7b: A functional organization structure will negatively moderate the influence of perceived ASD success on ASD routinization. Positive Influence H7c: A developmental organization culture will positively moderate the influence of perceived ASD success on ASD routinization. H7d: A projectized organization structure will positively moderate the influence of perceived ASD success on ASD routinization.
Proposition 8: ASD acceptance will not directly influence ASD routinization.	H8: ASD acceptance will not have a direct influence on ASD routinization.

Organizational culture

To investigate the relationship between o*rganizational culture* (*OC)* and post-adoptive ASD use further, we draw on the OC literature to highlight the composition of OC, types of OCs, and their effect on the use of ASD methods. OC is defined as "a pattern of shared basic assumptions learned by an [organization] as it solves its problems of external adaptation and internal integration, which has worked well enough to be considered valid and, therefore, to new members as the correct way to perceive, think, and feel in relation to those problems" (Schein 1985). In the IS literature, culture has been identified as an influencer of the use of information technology at the national, organizational, and group levels (Leidner and Kayworth 2006). Prior research has shown interest in the relationship between cultural values and the diffusion of IT (Kitchell 1995; Hoffman and Klepper 2000), and findings commonly suggest that different cultural orientations affect the propensity of IT adoption. Similarly, the extant literature suggests that the different cultural values were found to lead to different perceptions and approaches to software development (Leidner and Kayworth 2006). For instance, Iivari and Huisman's (2007) examination of the relationship between OC and the deployment of software development methods suggests that a hierarchical organizational culture orientation increases the deployment of these methods as perceived by IS developers and the rational organizational culture decreases it as perceived by IT managers. In relation to the integrated model in the thematic analysis section, OC combines the human and social factors (theme 4) and the organizational perspective (theme 2) to describe the human centric influences to ASD use at the organizational level. Therefore, OC is particularly

useful as a theoretical lens to examine the extent to which the social norms of an organization influence its ASD use.

In prior ASD literature, researchers have focused on the relationship between organizational culture and ASD use (Robinson and Sharp 2005, Tolfo and Wazlawick 2008, and Strode et al. 2009), where *use* is studied in either the early adoptive stages or conceptualized in the generic sense of use versus non-use, with few exceptions (Iivari and Iivari 2011). However, despite calls for future empirical investigation on the influence of organizational culture in deeper ASD usage scenarios (Chang and Thong 2009; Iivari and Iivari 2011, Vijayasarathy and Turk 2012), there still remains only a dearth of empirical examination on this relationship.

To understand this relationship, we concentrate on ASD as a form of process innovation (Chang and Thong 2009), often following traditional methods, that demands high levels of customer and stakeholder involvement throughout iterative development process. Agile's emphasis on social interactions and continual readiness towards changing requirements (Beck et al. 2001; Conboy 2009) has been shown to induce significant changes to more hierarchical organizational structures (Nerur 2005). As a result, the successful adoption of ASD methods implies changes to an organization's culture (Senapathi and Srinivasan 2012) in hierarchical culture scenarios. Furthermore, recent studies support the impact of OC as a critical factor in the ASD assimilation process, both in the ASD acceptance and routinization stages. For example, in a recent survey of 3925 respondents from the ASD community, the inability to change a company's OC ranked as

the leading inhibitor to ASD adoption (Version One 2015). In another recent survey, both the inability to change business culture, ranked first, and the inability to change IT culture, ranked third, were some of the most frequently cited inhibitors of ASD adoption across both initial and mature ASD teams (Ambler 2014). Drawing on prior research to explain the relationship between OC and both ASD acceptance and routinization, we propose:

> Proposition 1: A firm's organization culture, represented by its core values, will influence its acceptance of ASD methods.

In H1a and H1b, we hypothesize that different OC orientations, built upon different values will influence the adoption of ASD methods differently (Dubé 1998).

Organizational structure

Although organization culture's broad conceptualization makes it arguably connected to every organizational process, OC's complex, interrelated, and somewhat ambiguous set of factors make it impossible to create a comprehensive framework (Cameron and Quinn 2011). Consequently, researchers have had marginal success integrating and organizing elements of OC into widely used frameworks. One exemption is Schein's (1985) three layer model of basic assumptions, espoused values, and artifacts, which conceptualizes observable and unobservable layers of OC. His three layer model suggests that aspects of OC can be tapped by focusing on manifesting elements that are theoretically and practically tied to the unobservable layers such as espoused values (layer 2) and artifacts (layer 3). Thus, the vast majority of OC studies focus on values as constituents of OC

(Hofstede 1980; Quinn and McGrath 1985) because while values are not directly observable, they can be distilled from how people justify what they do (Schein 1985).

In this work, we draw upon the Competing Values Model (CVM) to conceptualize OC, following prior ISD research (Iivari and Huisman 2007). We use the CVM to ground our conceptualization of culture as the manifestation of a competing values system. These culture types form from four quadrants, differentiated by the continuums of two dimensions, internal to external focus and change to stability as shown in Figure 2.8 (Iivari and Huisman 2007).
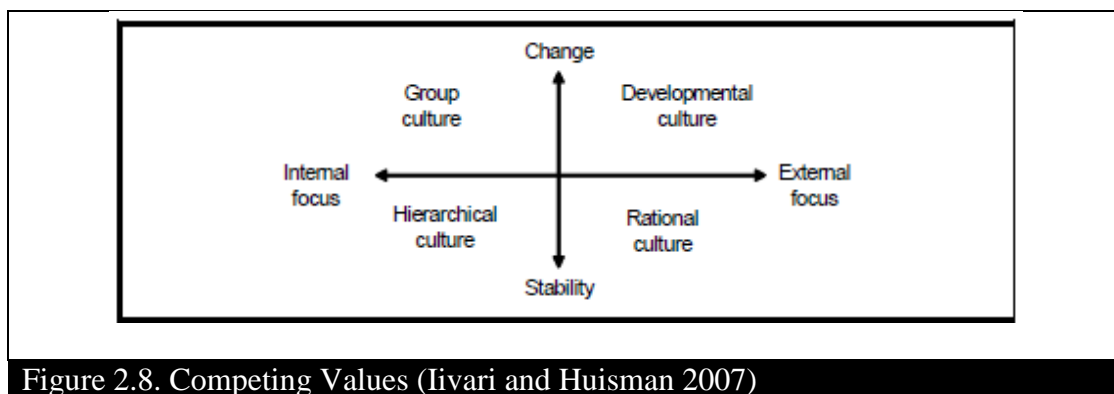
Figure 2.8. Competing Values (Iivari and Huisman 2007)

Each of the four quadrants in the CVM represents a distinct set of indicators of OC, which we will draw upon to formulate hypotheses regarding the use of ASD methods within a given set core of values. Using CVM, we propose that different types of cultures as represented by their set of core values, will produce organizational structures to support the cultural values. *Organizational structure (OS)* is defined as "the logically consistent clustering of an organization's elements as it searches for harmony in its internal processes and consonance with its environment" (Mintzberg 1980). Thus, organizational structures provide mechanisms through which organizations communicate

explicitly and implicitly about their assumptions. In reference to ASD methods, we make the following propositions:

Proposition 2: A firm's organizational culture, represented by its core values, will influence its organization structure (form).

Proposition 3: A firm's organizational structure will influence its acceptance of ASD methods.

Prior research has emphasized the importance of cultural compatibility in relationship to a given IS effort (Dubé 1998). That withstanding, the compatibility and relative fit of a top-down, hierarchical organizational culture to a bottom-up methodology like agile, has been scrutinized (Boehm and Turner 2005; Nerur and Balijepally 2007). Given proposition 2, we also propose that a hierarchical organizational culture (Quinn 1988) predicated on bureaucracy, will produce organizational structures to support the culture (H2a). On the other hand, we propose that developmental organizational cultures (Quinn 1988) predicated on adhocracy, will produce organizational structures to support the culture. Prior research suggests that these bottom-up cultures are more compatible with ASD methods (Iivari and Iivari 2011), thus our formulation of H2b.

In our research, we focus specifically on hierarchical and developmental organizational cultures because our primary proposition is that ASD values compete and conflict with those of a hierarchical organizational culture. Thus, a hierarchical organizational culture should have a negative influence on ASD routinization. Given that the hierarchical and

developmental organizational cultures are polar opposites in the CVM framework, these culture types pose competing and conflicting demands on organizations (Iivari and Huisman 2007). Second, from a competing values standpoint, the emphasis of the developmental organizational culture closely aligns with the tenets of the ASD philosophy as espoused in the Agile Manifesto. Thus, we chose a more parsimonious approach to test our hypothesis by focusing on the hierarchical and developmental organizational cultures to discriminate between organizational cultures based on their most salient differences.

Table 2.8 summarizes that difference between the hierarchical and developmental organizational cultures, and their resulting organizational structures based on the prior research of Russo, Shams, and Fitzgerald (2013), Iivari and Huisman (2007), and Gallivan (2001).

Table 2.8. A Comparison of Hierarchical and Developmental Cultures		
Organizational Culture	Hierarchy	Developmental
Organizational Strategy	Specified and predictable	Flexibility and adaptive
Enterprise Governance Structure	Bureaucratic	Adhocracy
Team Management Structure	Functional	Projectized
Decision Making Structure	Top-down	Bottom-up
Team Work Structure	Siloed	Collaborative
Control Mode	Formal	Informal
Organizational Focus	Internal	External
Stability vs. Change	Stability	Change

Organizational Culture Orientation to ASD Acceptance, Project Success, and ASD Routinization

ASD Acceptance and Perceived Project Success

ASD acceptance refers to the introductory adoption and employment of ASD methods as a process innovation for one or more software development projects (Vijayasarathy and Turk 2012; Wang et al. 2012). Both the increase in ASD adoption rates and the findings of empirical research suggest that the use of ASD methods yields a number of comparative advantages over traditional methods such as higher team productivity (Layman et al. 2004), higher customer satisfaction (Ceschi et al. 2005), more efficient requirements gathering (Hansen and Lyytinen 2014), and a greater ability to handle changing user requirements (Vigden and Wang 2009), with few studies reporting contrary results (Dalcher et al. 2005). Furthermore, the use of ASD methods has been associated with a higher rate of project success and in turn, a lower rate of project failure (Chaos 2010). From an organizational perspective, the success of ASD projects has been determined by not only traditional factors of time, scope, and budget (Keider 1974; Boehm, 1981; Pinto and Slevin 1988), but also factors such as meeting user requirements (Procaccino and Verner 2006).

As previously, noted, extant literature suggests that different cultural values were found to lead to different perceptions and approaches to software development (Leidner and Kayworth 2006). Therefore, we propose that different organization factors, culture and structure, will have significant influences on both of ASD acceptance and perceived project success (Jiang, Klein, and Pick 2003), and we hypothesize these influences according to the differing influence of hierarchal versus developmental organizational

cultures on ASD use (H1a, H1b, H3a, H3b, H4a, H4b, H4c, and H4d). In addition, drawing on previous findings (Ceschi et al. 2005; Karlström and Runeson 2006; Hummel and Rosenkranz 2013), we hypothesize a positive relationship between ASD use and perceived project success (H5).

Proposition 4: Organizational factors, culture and structure, will influence perceived ASD success.

Proposition 5: A firm's ASD acceptance will influence its perceived ASD success.

ASD Routinization

ASD routinization refers to the usage of ASD methods as a normal activity in the organization (Wang et al. 2012). Prior research suggests that the impacts of using a particular software development methodology will influence developers' attitudes towards the future use of the methodology (Khalifa and Verner 2000). Prior research also suggests that developers may be more apt to use a methodology if he or she perceives that they are more effective when using the methodology (Green and Hevner 1999). Thus, it follows that a firm's perceived ASD success during ASD acceptance will positively influence its attitudes toward ASD routinization.

Proposition 6: A firm's perceived ASD success will influence ASD routinization.

On the other hand, some organizations do not routinize ASD methods even after experiencing benefits of agile use (Sutherland 2014). In these cases, the decision

concerning whether an organization continues its use of ASD methods is determined by compatibility of the methodology with the organizational culture and social norms (Dubé and Robey 1999). Prior research points out that change to both the organization and the innovation may occur in order to exploit the innovation (Rogers 1983; Gallivan 2001). During the implementation phase, an innovation may conflict with sources of structure and social norms within an organization, which may lead to the rejection of a particular innovation. On the other hand, when ASD acceptance leads to perceive ASD success, this relationship may be assumed to lead to the routinization of ASD methods in all cases. However, we propose that organizational factors, particularly culture and structure, significantly influence the relationship between perceived ASD success and ASD routinization. Thus,

> Proposition 7: Organizational factors, culture and structure, will influence ASD routinization beyond the perceived success of ASD acceptance.

Furthermore, the process of adapting an innovation includes redefining and restructuring organizational processes, so that members understand the innovation and its role in meeting organizational goals (Gallivan 2001). Therefore, we hypothesize that when organizational values are in competition with agile values, organizational factors of culture and structure may inhibitor the assimilation of ASD methods (7a, 7b). On the other hand, when organizational values align with ASD values, organizational factors, culture and structure, may facilitate the assimilation of ASD methods (7c, 7d). These hypotheses challenge the notion that successful initial agile use alone leads to routine use (H8). Thus,

Proposition 8: ASD acceptance will not directly influence ASD routinization.

RESEARCH CHALLENGES (LIMITATIONS)

As set forth in prior research, defining culture poses many challenges to researchers. Although culture has commonly been conceptualized as consisting of multiple levels, organizational culture is still difficult to define, conceptualize, and operationalize with certainty. Our work follows prior research, which draws upon values while accepting that there are other ways to study culture.

In our study, we aim to develop a parsimonious model to explicate the impact of key organizational factors on ASD use. As a consequence, our model does not elaborate on the recursive relationships that may exists between constructs. The extant literature in both themes 2 and 4 conceptualize factors that influence use, but these themes rarely discuss how use impacts these factors in reciprocally. Future research is needed to further investigate these mutual impacts.

We have done our best to be thorough in our literature search (see Appendix A9), however, we are aware that some articles may have been missed. What we hoped to achieve is a theory building exercise that moves closer to a unified understanding of ASD. Future research may highlight contributions from articles not contained in this review.

CONCLUSION

This systematic review set forth to assess the theoretical perspectives in ASD research. This review performed a thorough search of the literature and clearly answered four important research questions. This study has contributed to the literature base by identifying the theoretical perspectives called for in previous research, synthesizing the current body of knowledge, highlighting gaps in the literature, and proposing a new model to guide future research.

First, the theoretical assessments relating to the current body of research have been identified (see Appendix A6). Second, the synthesis of these theoretical assessments contributes toward the development of a unified framework for understanding the ASD research as mentioned in previous reviews on ASD research (Dybå and Dingsøyr 2008, Dingsøyr et al. 2012) (see Figure 2.6). Third, the framework for organizing theoretical assessments in empirical research aids researchers in understanding the current gaps in our theoretical understanding of ASD (see Table 2.4 and Appendix A8). Fourth, and predominantly, our model on ASD routinization investigates a specific assimilation gap, extending the current state of knowledge about ASD methodology assimilation in organizations. Table 2.4 and Appendix A8 point out major gaps in our understanding of ASD assimilation at both the organizational level and post-acceptance stages. This research extends prior investigations on the relationship between OC and software development (Iivari and Huisman 2007) by clarifying the role of organizational factors of culture and structure in the routinization of ASD methods. This pivotal finding is

accentuated in our model, which provides opportunities for post-acceptance research on ASD assimilation.

During this study, we encountered two main approaches to studying ASD from a theoretical perspective, one on ASD as a process, and the other on ASD as a process innovation. Our synthesis of Themes 1 through 4 helps to show that the ASD phenomenon has an organization-wide impact. Thus, ASD research should consider the impacts of ASD assimilation in areas that remain understudied (see Appendix A8). Given that our final model primarily focuses on ASD as a process innovation with organization-wide impacts, future research may explore other organizational level notions that may aid in our understanding ASD assimilation such as organizational learning and organizational memory. These factors may have a profound effect on the way collectives approach ASD methods and cognitively process ASD assimilation activities in specific organizational settings.

Another opportunity for future research is in the areas of scale development and construct validation. Currently, there are few validated surveys on ASD. Findings from this study highlight the impact of ASD on multiple organizational levels involved in the systems development process, which may suggest that ASD be measured as a multi-level construct. Additionally, ASD constructs should represent unique components of ASD development such as iterative development and self-organization.

Although our theoretical model was developed to study ASD, future research may explore the application of this model for other forms of innovation. Further research could examine to what extent different contexts affect the theory. Such research can demonstrate either the uniqueness (to ASD) or broad application of this theoretical model.

The practical contributions of this research are listed as follows. First, our model sets forth a set of propositions that can aid organizations in understanding ASD routinization issues. Particularly, the impact of an organization's culture on specific factors can be gained from this research. Whereas past research and practitioner surveys have proposed the role of organizational culture in the ASD process, there is little research that has sought to explain how organizational culture might affect the routinization process. Once operationalized, our model may break new ground in the area of ASD assimilation research. Furthermore, understanding ASD routinization holds significant impact for understanding agile transformation, or the organizational change from non-agile to agile, as the ASD assimilation process is key in an organization's pursuit of firm agility.

Second, the history of ASD research has followed pertinent issues from the ASD practice, beginning with developing a better way to develop software to current issues such as how to optimize ASD processes in organizational settings. Our research themes record the theoretical contributions chronologically, which highlight how ASD practices have been studied throughout the years. In addition, since ASD research has followed the ASD practice, this chronology reveals a brief account of practice-related questions.

In conclusion, the authors used a systematic process to make this review both rigorous and replicable. Because this research is one of the initial efforts to provide a multidisciplinary review of empirically and theoretically grounded studies in ASD research, we hope that this study will serve as a great help to current and future researchers of ASD processes.

REFERENCES

Abrahamsson, P., Conboy, K., and Xiaofeng, W. 2009. "'Lots done, more to do': the current state of agile systems development research," Editorial, *European Journal of Information Systems,* pp. 281-284.

Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J. 2002. "Agile software development methods," *VTT publications,* (478:3), pp. 167-168.

Acuña, S.T., Gómez, M. and Juristo, N. 2009. "How do personality, team processes and task characteristics relate to job satisfaction and software quality?," *Information and Software Technology*, (51:3), pp. 627-639.

Adolph, S., Kruchten, P. and Hall, W., 2012. "Reconciling perspectives: A grounded theory of how people manage the process of software development," *Journal of Systems and Software*, (85:6), pp. 1269-1286.

Ågerfalk, P.J., Fitzgerald, B. and Slaughter, S.A., 2009. "Introduction to the Special Issue-Flexible and Distributed Information Systems Development: State of the Art and Research Challenges," *Information systems research*, (20:3), pp.317-328.

Alaa, G. and Fitzgerald, G., 2013. "Re-conceptualizing agile information systems development using complex adaptive systems theory," *Emergence: Complexity and Organization*, (15:3), p.1.

Ambler, S. 2007. "Agile Adoption Rate Survey Results: March 2007." Survey. Accessed online. 11 August 2015.

Ambler, S. 2014. "2014 Agile Adoption Rate Survey Results." Survey. Accessed online. 11 June 2015.

Babb, J., Hoda, R. and Norbjerg, J. 2014. "Embedding reflection and learning into agile software development," *IEEE software*, (4), pp. 51-57.

Bahli, B. and Zeid, E.A. 2005. "The role of knowledge creation in adopting extreme programming model: an empirical study," In *Information and Communications Technology. Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference on*, pp. 75-87.

Balijepally, V., Mahapatra, R., and Nerur, S. 2006. "Assessing Personality Profiles of Software Developers in Agile Development Teams," *Communications of AIS* (18:4), pp. 2-40.

Barlow, J. B., Giboney, J. S., Keil, M. J., Wilson, D. W., Schuetzler, R. M., Lowry, P. B., and Vance, A. 2011. "Overview and Guidance on Agile Development in Large Organizations," *Communications of AIS* (29:2), pp. 25-44.

Baskerville, R., Pries-Heje, J., and Madsen, S. 2011. "Post-agility: What Follows a Decade of Agility?," *Information and Software Technology* (53:5), pp. 543-555.

Baskerville, R., Ramesh, B., Levine, L., Pries-Heje, J., and Slaughter, S**.** Sandra 2003. "Is Internet-Speed Software Development Different?," *IEEE Software* (20:6), pp. 70-77.

Baskerville, R. and Wood-Harper, A.T., 1998. "Diversity in information systems action research methods," *European Journal of information systems*, (7:2), pp.90-107.

Batra, D., VanderMeer, D. and Dutta, K. 2011. "Extending agile principles to larger, dynamic software projects: a theoretical assessment," *Journal of Database Management (JDM)*, (22:4), pp. 73-92.

Beck et al. 2001. Agile Alliance, Agile Manifesto.

Bishop, D., and Deokar, A. 2014. "Toward an Understanding of Preference for Agile Software Development Methods from a Personality Theory Perspective," *System Sciences (HICSS), 2014 47th Hawaii International Conference on,* IEEE, pp. 4749-4758.

Boehm, B. 1981. *Software engineering economics*, (Prentice-hall Englewood Cliffs (NJ).

Boehm, B., and Turner, R. 2005. "Management challenges to implementing agile processes in traditional development organizations," *Software, IEEE* (22:5), pp. 30-39.

Boschetti, M.A., Golfarelli, M., Rizzi, S. and Turricchia, E. 2014. "A Lagrangian heuristic for sprint planning in agile software development," *Computers and Operations Research*, *43*, pp. 116-128.

Cameron, K. S. and Quinn, R. E. 2011. *Diagnosing and changing organizational culture: Based on the competing values framework*. 3rd ed., John Wiley and Sons.

Cao, L. 2004. "Modeling dynamics of agile software development," in *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, ACM, October, pp. 46-47.

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

Ceschi, M., Sillitti, A., Succi, G., and Panfilis, S. D. 2005. "Project management in plan-based and agile companies," *Software, IEEE* (22:3), pp 21-27.

Chan, F. K., and Thong, J. Y. 2009. "Acceptance of agile methodologies: A critical review and conceptual framework," *Decision Support Systems* (46:4), pp. 803-814.

Chaos, tech. report, Standish Group Int'l, 2010.

Cohen, D., Lindvall, M. and Costa, P. 2004. "An introduction to agile methods," *Advances in computers*, 62, pp.1-66.

Conboy, K. 2009. "Agility from first principles: reconstructing the concept of agility in information systems development," *Information Systems Research* (20:3), pp. 329-354.

Conboy, K. and Duarte, V. 2010. "Scaling Agile to Lean–Track Summary," in *Lean Enterprise Software and Systems,* Springer Berlin Heidelberg, pp. 1-2.

Conboy, K., Pikkarainen, M., and Wang, X. 2007. "Agile Practices in Use from an Innovation Assimilation Perspective: A Multiple Case Study," *ICIS 2007 Proceedings.* Paper 7.

Cooper, R. B., and Zmud, R. W. 1990. "Information technology implementation research: a technological diffusion approach," Management science (36:2), pp. 123-139.

Critical Appraisal Skills Programme (CASP). Retrieved November 10, 2012, from http://www.casp-uk.net/.

Dalcher, D., Benediktsson, O., and Thorbergsson, H. 2005. "Development life cycle management: a multiproject experiment," in *Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the, IEEE,* pp. 289-296.

Denning, S. 2011. "Successfully implementing radical management at Salesforce.com," *Strategy and Leadership* (39:6), pp. 4-10.

Dennis, A., and Valacich, J. 2001. "Conducting research in information systems," *Communications of the AIS,* (7:5), pp. 1-41.

Dhillon, G., 1997. *Managing information system security*. London, UK, Macmillan Press.

Dingsøyr, T., and Hanssen, G.K. 2002. "Extending agile methods: postmortem reviews as extended feedback," in *Advances in Learning Software Organizations*, Springer Berlin Heidelberg, pp. 4-12.

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B. 2012. "A decade of agile methodologies: Towards explaining agile software development," *Journal of Systems and Software* (85:6), pp. 1213-1221.

Drury, M., Conboy, K. and Power, K. 2012. "Obstacles to decision making in Agile software development teams," *Journal of Systems and Software*, (85:6), pp. 1239-1254.

Dubé, L. 1998. "Teams in packaged software development: The software corp. Experience," *Information Technology and People*, (11:1), pp. 36-61.

Dubé, L., and Robey, D. 1999. "Software stories: three cultural perspectives on the organizational practices of software development," *Accounting, Management and Information Technologies*, (9:4), pp. 223-259.

Dybå, T., and Dingsøyr, T. 2008. "Empirical studies of agile software development: A systematic review.," *Information and software technology* (50:9), pp. 833-859.

Erickson, J., Lyytinen, K., and Siau, K. 2005. "Agile modeling, agile software development, and extreme programming: the state of research," *Journal of Database Management (JDM)* (16:4), pp 88-100.

Fichman, R. G. and Kemerer, C. F. 1999. "The Illusory Diffusion of Innovation: An Examination of Assimilation Gaps," *Information systems research*, (10:3), pp. 255-275.

Fink, A. 2005. *How to conduct surveys: A step-by-step guide*: Sage.

Gallivan, M. J. 2001. "Organizational adoption and assimilation of complex technological innovations: development and application of a new framework," *ACM Sigmis Database* (32:3), pp. 51-85.

Ghobadi, S. and Mathiassen, L. 2015. "Perceived barriers to effective knowledge sharing in agile software teams," *Information Systems Journal*.

Goh, J.C.L., Pan, S.L. and Zuo, M., 2013. Developing the agile IS development practices in large-scale IT projects: the trust-mediated organizational controls and IT project team capabilities perspectives. *Journal of the Association for Information Systems*, *14*(12), p.722.

Gregory, T., Mathiassen, L. and Sambhara, C. 2013. "Chains of control in agile software development," in *AMCIS*.

Hansen, S. and Lyytinen, K., 2014. "Requirements Computation: Analyzing Requirements Cognition in Multiple Development Paradigms," in *AMCIS*.

Harris, M. L., Collins, R. W., and Hevner, A. R. 2009a. "Control of Flexible Software Development Under Uncertainty," *Information Systems Research* (20:3), pp. 400-419.

Harris, M. L. Hevner, A. R., and Collins, R. W. 2009b. "Controls in Flexible Software Development," *Communications of AIS* (24), pp. 757-776.

Hilkka, M. R., Tuure, T., and Matti, R. 2005. "Is extreme programming just old wine in new bottles: a comparison of two cases," *Journal of Database Management (JDM)* (16:4), pp 41-61.

Hoda, R., Noble, J., and Marshall, S. 2011. "The Impact of Inadequate Customer Collaboration on Self-organizing Agile Teams," *Information and Software Technology* (53:5), pp. 521-534.

Hoda, R., Noble, J., and Marshall, S. 2013. "Self-Organizing Roles on Agile Software Development Teams," *IEEE Transactions on Software Engineering* (39:3), pp. 422-444.

Hoffman, N., and Klepper, R. 2000. "Assimilating new technologies: The role of organizational culture," *Information Systems Management* (17:3), pp. 36-42.

Hofstede, G. 1980. *Culture's Consequences: International Differences in Work-Related Values*, Sage Publications, Beverly Hills, CA, pp. 335-355.

Hollis, B., and Maiden, N. 2013. "Extending agile processes with creativity techniques," *Software, IEEE*, (30:5), pp. 78-84.

Hong, W., Thong, J. Y., Chasalow, L. C., and Dhillon, G. 2011. "User acceptance of agile information systems: A model and empirical test," *Journal of Management Information Systems* (28:1), pp 235-272.

Hummel, M. 2014. "State-of-the-Art: A Systematic Literature Review on Agile Information Systems Development," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pp. 4712-4721.

Hummel, M., Rosenkranz, C., and Holten, R. 2013. "Explaining The Changing Communication Paradigm Of Agile Information Systems Development: A Research Model, Measurement Development And Pretest," in *Proceedings of the 21st European Conference on Information Systems*, p. 17.

Iivari, J., and Huisman, M. 2007. "The Relationship Between Organizational Culture and the Deployment of Systems Development Methodologies," *MIS Quarterly*, pp 35-58.

Iivari, J. and Iivari, N. 2011. "The Relationship Between Organizational Culture and the Deployment of Agile Methods," *Information and Software Technology* (53:5), pp 509-520.

Jiang, J. J., Klein, G., and Pick, R. A. 2003. "The impact of IS department organizational environments upon project team performances," *Information and Management* (40:3), pp. 213-220.

Karlsson, F., and Ågerfalk, P. 2009. "Towards structured flexibility in information systems development: Devising a method for method configuration," *Journal of Database Management (JDM)* (20:3), pp. 51-75.

Karlström, D., and Runeson, P. 2006. "Integrating agile software development into stage-gate managed product development," *Empirical Software Engineering* (11:2), pp 203-225.

Karlström, D., Runeson, P. and Norden, S. 2005. "A minimal test practice framework for emerging software organizations," *Software Testing, Verification and Reliability*, (15:3), pp. 145-166.

Katerattanakul, P., and Han, B. 2003. "Are European IS Journals under-rated? An answer based on citation analysis," *European Journal of Information Systems,* (12:1), p. 60.

Keider, S. P. 1974. "Why projects fail," *Datamation* (20:12), pp 53-55.

Keith, M., Demirkan, H., and Goul, M. 2013. "Service-oriented methodology for systems development," *Journal of Management Information Systems*, (30:1), pp. 227-260.

Kitchell, S. 1995. "Corporate culture, environmental adaptation, and innovation adoption: a qualitative/quantitative approach," *Journal of the Academy of Marketing Science* (23:3), pp. 195-205.

Kitchenham, B., and Charters, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*.

Kwon, T.H. and Zmud, R.W., 1987. "Unifying the fragmented models of information systems implementation," in *Critical issues in information systems research*, John Wiley and Sons, Inc., pp. 227-251.

Layman, L., Williams, L., and Cunningham, L. 2004. "Exploring extreme programming in context: an industrial case study," *Agile Development Conference, 2004, IEEE*, pp. 32-41.

Layman, L., Williams, L., Slaten, K., Berenson, S. and Vouk, M. 2008. "Addressing diverse needs through a balance of agile and plan-driven software development methodologies in the core software engineering course," *International Journal of Engineering Education*, (24:4), p. 659.

Lee, G., DeLone, W., and Espinosa, J. A. 2006. "Ambidextrous Coping Strategies in Globally Distributed Software Development Projects," *Communications of the ACM* (49:10), pp. 35-40.

Lee, G., and Xia, W. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data," *MIS Quarterly* (34:1), pp 87-114.

Leidner, D.E. and Kayworth, T., 2006. "Review: a review of culture in information systems research: toward a theory of information technology culture conflict," *MIS quarterly*, (30:2), pp.357-399.

Lévárdy, V. and Browning, T.R. 2009. "An adaptive process model to support product development project management," *Engineering Management, IEEE Transactions on*, (56:4), pp. 600-620.

Lowry, P. B., Romans, D., and Curtis, A. 2004. "Global Journal Prestige and Supporting Disciplines: A Scientometric Study of Information Systems Journals," *Journal of the Association for Information Systems,* (5:2), pp. 29-77.

Mangalaraj, G., Mahapatra, R. and Nerur, S. 2009. "Acceptance of software process innovations–the case of extreme programming," *European Journal of Information Systems*, (18:4), pp. 344-354.

Markus, M. L. and Robey, D. "The organizational validity of management information systems," *Human relations,* (36:3), pp. 203-225.

Maruping, L. M., Venkatesh, V., and Agarwal, R. 2009a. "A Control Theory Perspective on Agile Methodology Use and Changing User Requirements," *Information Systems Research* (20:3), pp. 377-399.

Maruping, L. M., Zhang, X., and Venkatesh, V. 2009b. "Role of Collective Ownership and Coding Standards in Coordinating Expertise in Software Project Teams," *European Journal of Information Systems* (18:4), pp 355-371.

McAvoy, J. and Butler, T., 2007. "The impact of the Abilene Paradox on double-loop learning in an agile team," *Information and software technology*, (49:6), pp. 552-563.

McAvoy, J. and Butler, T. 2009. "The role of project management in ineffective decision making within Agile software development projects," *European Journal of Information Systems*, (18:4), pp. 372-383.

McAvoy, J., Nagle, T., and Sammon, D. 2013. "Using mindfulness to examine ISD agility," *Information Systems Journal* (23:2), pp. 155-172.

McBreen, P., 2003. *Questioning Extreme Programming*. Addison-Wesley.

Melo, C.D.O., Cruzes, D.S., Kon, F. and Conradi, R., 2013. "Interpretative case studies on agile team productivity and management," *Information and Software Technology*, (55:2), pp. 412-427.

Meso, P., and Jain, R. 2006. "Agile Software Development: Adaptive Systems Principles and Best Practices," *Information Systems Management* (23:3) Summer2006, pp. 19-30.

Meyer, A. D. and Goes, J. B. 1988. "Organizational assimilation of innovations: A multilevel contextual analysis," *Academy of management journal* (31:4), pp. 897-923.

Mintzberg, H. 1980. "Structure in 5's: A Synthesis of the Research on Organization Design," *Management science* (26:3), pp. 322-341.

"MIS Journal Rankings." *AIS Electronic Library AISeL.* Retrieved July 20, 2014.

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*, (52:5), pp. 480-491.

Mumford, E. 1983. *Designing human systems for new technology: the ETHICS method.* Manchester: Manchester Business School.

Mylonopoulos, N. A., and Theoharakis, V. 2001. "Global Perceptions of IS Journals," *Communications of the ACM,* (44:9), pp. 29-33.

Nerur, S., and Balijepally, V. 2007. "Theoretical reflections on agile development methodologies," *Communications of the ACM*, (50:3), pp.79-83.

Nerur, S., Mahapatra, R., and Mangalaraj, G. 2005. "Challenges of migrating to agile methodologies," *Communications of the ACM* (48:5), pp 72-78.

Northover, M. 2008. "Towards a Philosophical Understanding of Agile Software Methodologies," PhD diss., University of Pretoria.

Northover, M., Boake, A. and Kourie, D.G., 2006. "Karl Popper's critical rationalism in agile software development," In *Conceptual Structures: Inspiration and Application*, Springer Berlin Heidelberg, pp. 360-373.

Okoli, C., and Schabram, K. 2010. "A guide to conducting a systematic literature review of information systems research," *Sprouts Work. Pap. Inf. Syst*, (10), p. 26.

Overhage, S., and Schlauderer, S. 2012. "Investigating the long-term acceptance of agile methodologies: An empirical study of developer perceptions in scrum projects," in the *45th Hawaii International Conference on System Sciences (HICSS),* pp. 5452-5461.

Parolia, N., Goodman, S., Li, Y. and Jiang, J.J., 2007. "Mediators between coordination and IS project performance," *Information & Management*, (44:7), pp. 635-645.

Peffers, K., and Ya, T. 2003. "Identifying and evaluating the universe of outlets for information systems research: Ranking the journals," *Journal of Information Technology Theory and Application,* (5:1), pp. 63-84.

Persson, J. S., Mathiassen, L., and Aaen Ivan 2012. "Agile distributed software development: enacting control through media and context," *Information Systems Journal* (22:6), pp. 411-433.

Petticrew, M., and Roberts, H. 2006. *Systematic reviews in the social sciences: A practical guide*: Wiley.com.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., and Still, J. 2008. "The impact of agile practices on communication in software development," *Empirical Software Engineering* (13:3), pp. 303-337.

Pinto, J. K., and Slevin, D. P. 1988. "Project success: definitions and measurement techniques," Project Management Institute.

Port, D. and Bui, T., 2009. "Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications," *European Journal of Information Systems*, (18:4), pp. 317-331.

Procaccino, J. D., and Verner, J. M. 2006. "Software project managers and project success: An exploratory study," *Journal of Systems and Software* (79:11), pp. 1541-1551.

Quinn, R. E. 1988. *Beyond Rational Management: Mastering the Paradoxes and Competing Demands of High Performance*, Jossey-Bass, San Francisco.

Quinn, R. E., and McGrath, M. R. 1985. "The Transformation of Organizational Cultures: A Competing Values Perspective" in *Organizational Culture*, P. J. Frost, L. F. Moore, M. R. Louis, C. C. Lundberg, and J. Martin (eds.), Sage Publications, London, pp. 314-334.

Rainer Jr., R., and Miller, M. 2005. "Examining differences across journal rankings," *Communications of the ACM,* (48:2), pp. 91-94.

Ridley, D. 2008. *The literature review: A step-by-step guide for students*: Sage.

Robey, D., Im, G. and Wareham, J.D., 2008. "Theoretical foundations of empirical research on interorganizational systems: assessing past contributions and guiding future directions," *Journal of the Association for Information Systems*, (9:9), p.497.

Robinson, H., and Sharp, H. 2005. "Organisational culture and XP: three case studies," in *Proceedings of the 2005 Agile Conference*, IEEE, pp. 49-58.

Rogers, E. 1983. "Diffusion of innovations," *New York: Free Press,* (18:20), pp. 271.

Rogers, E. M. 1962. *Diffusion of innovations*. Simon and Schuster.

Rogers, E. M. 1983. *Diffusion of Innovations*, 3rd ed., New York: Free Press.

Russo, N. L., Fitzgerald, G. and Shams, S., 2013. "Exploring adoption and use of agile methods: A comparative case study," in *AMCIS*.

Ryan, S. and O'Connor, R.V. 2013. "Acquiring and sharing tacit knowledge in software development teams: An empirical study," *Information and Software Technology*, (55:9), pp. 1614-1624.

Santos, V., Goldman, A., Martins, D. and Cortes, M. 2014. "The Influence of Organizational Factors on Inter-team Knowledge Sharing Effectiveness in Agile Environments," in the *47th Hawaii International Conference on System Sciences (HICSS)*, pp. 4729-4738.

Scheerer, A., Hildenbrand, T. and Kude, T. 2014. "Coordination in large-scale agile software development: A multiteam systems perspective," In *System Sciences (HICSS), 2014 47th Hawaii International Conference on,* pp. 4780-4788.

Schein, E. H. 1985. *Organizational Culture and Leadership*, Jossey-Bass, San Francisco, CA.

Schmidt, C., Kude, T., Heinzl, A. and Mithas, S. 2014. "How Agile practices influence the performance of software development teams: The role of shared mental models and backup," *in Proceedings of the International Conference on Information Systems.*

Senapathi, M., Drury, M. and Srinivasan, A. 2013. "Agile Usage: Refining a Theoretical Model," in *PACIS*, p. 43.

Senapathi, M., and Srinivasan, A 2012. "Understanding post-adoptive agile usage: An exploratory cross-case analysis," *Journal of Systems and Software* (85:6), pp 1255-1268.

Senapathi, M. and Srinivasan, A. 2013. "Sustained agile usage: a systematic literature review," in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, pp. 119-124.

Senapathi, M., and Srinivasan, A. 2014. "An Empirical Investigation of the Factors Affecting Agile Usage," *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ACM, p. 10.

Sharp, H., and Robinson, H., 2008. "Collaboration and co-ordination in mature eXtreme programming teams," *International Journal of Human-Computer Studies*, (66:7), pp. 506-518.

Socha, D., and Walter, S. 2007. "Is designing software different from designing other things?," *International Journal of Engineering Education*, (22:3), p. 540.

Strode, D. E., Huff, S. L., Hope, B., and Link, S. 2012. "Coordination in co-located agile software development projects," *Journal of Systems and Software* (85:6), pp 1222-1238.

Strode, D. E., Huff, S. L., and Tretiakov, A. 2009. "The impact of organizational culture on agile method use," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, IEEE, pp. 1-9.

Sutherland, J. 2014. *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Business.

Svensson, H. and Höst, M. 2005. "Introducing an agile process in a software maintenance and evolution organization," In *Software Maintenance and Reengineering, 2005. CSMR 2005. Ninth European Conference on*, pp. 256-264.

Tessem, B. 2003. "Experiences in learning XP practices: A qualitative study," In *Extreme Programming and Agile Processes in Software Engineering,* Springer Berlin Heidelberg, pp. 131-137.

Tolfo, C., and Wazlawick, R. S. 2008. "The Influence of Organizational Culture on the Adoption of Extreme Programming," *Journal of systems and software* (81:11), pp 1955-1967.

Trinidad, P., Benavides, D., Durán, A., Ruiz-Cortés, A. and Toro, M., 2008. "Automated error analysis for the agilization of feature modeling," *Journal of Systems and Software*, (81:6), pp. 883-896.

Version One. 2014. "8th Annual State of Agile Survey." Survey. Accessed online. 1 February 2015.

Version One. 2015. "9th Annual State of Agile Survey." Survey. Accessed online. 15 May 2015.

Vidgen, R., and Wang, X. 2009. "Coevolving systems and the organization of agile software development," *Information Systems Research* (20:3), pp. 355-376.

Vijayasarathy, L., and Turk, D. 2012. "Drivers of agile software development use: Dialectic interplay between benefits and hindrances," *Information and Software Technology* (54:2), pp 137-148.

Wagner, E.L., Newell, S. and Ramiller, N. 2013. "The structuring of freedom in agile development," in *AMCIS*.

Wang, X., Conboy, K., and Pikkarainen, M. 2012. "Assimilation of agile practices in use," *Information Systems Journal* (22:6), pp. 435-455.

Webster, J., and Watson, R. T. 2002. "Analyzing the Past to Prepare for the Future: Writing a Literature Review," *MIS Quarterly, 26*, xiii-xxiii.

Weisstein, E. W. 2012. Control Theory. from http://mathworld.wolfram.com/ControlTheory.html.

Whitman, M. E., Hendrickson, A. R., and Townsend, A. M. 1999. "Research Commentary. Academic Rewards for Teaching, Research, and Service: Data and Discourse," *Information Systems Research,* (10:2), pp. 99-109.

Wood, S., Michaelides, G., and Thomson, C. 2013. "Successful extreme programming: Fidelity to the methodology or good teamworking?," *Information and Software Technology* (55:4), pp. 660-672.

Yadav, V., Adya, M., Sridhar, V., and Nath, D. 2009. "Flexible Global Software Development (GSD): Antecedents of Success in Requirements Analysis," *Journal of Global Information Management,* (17:1), pp. 1-31.

Yu, X. and Petter, S. 2014. "Understanding agile software development practices using shared mental models theory," *Information and Software Technology*, (56:8), pp. 911-921.

CHAPTER 3: BRIDGING THE ACCEPTANCE-ROUTINIZATION GAP IN AGILE SOFTWARE DEVELOPMENT ASSIMILATION: AN EXPLORATORY CROSS-CASE ANALYSIS

INTRODUCTION

The *agile software development* (ASD) practice can be defined as a software development team's ability to respond to changing user requirements through a process of continual readiness (Conboy 2009; Lee and Xia 2010). ASD methods represent a departure from the heavily regimented and document-driven procedures of traditional, waterfall approaches to building software. In contrast to traditional approaches, ASD methods focus on adapting quickly to changing user requirements and to using less time for documentation in order to build working software quickly and iteratively through a collaborative effort. Reported findings in both practitioner media and academic research suggest that the use of ASD methods often lead to information systems development (ISD) process improvements (Drury et al. 2012; Strode et al. 2012; Daneva 2013), which in turn yield positive ISD outcomes such as faster times to market, higher software quality, and higher customer satisfaction when compared to waterfall methods (Cao et al. 2009; Overhage and Schlauderer 2012). The recognition of these and other benefits has led to the further move to adopt ASD methods by organizations that engage in ISD, both for internal users or external clients. In fact, ASD adoption rates both within organizations and across the ISD industry have seen significant growth since the signing of the *Manifesto for Agile Software Development* in 2001 (Beck et al. 2001). For instance, one survey reports an increase in Microsoft's ASD adoption rate from 34% in

2006 to 57% in 2012 (Murphy et al. 2013), and another survey reports that companies that plan to implement ASD for future development projects increased from 59 percent in 2011 to 83 percent in 2012 (Version One 2013). Despite growth in the overall adoption rates over the past two decades, many organizations are not realizing the promised benefits of the routine use of ASD methods (Denning 2012a, 2012b, Version One 2015). Indeed although some ASD adopters do achieve the benefits of ASD methods beyond initial ASD usage scenarios, many organizations have elected not to implement ASD methods for future development projects. This begs the question: why are some organizations not using ASD methods routinely despite their promised and reported benefits?

To answer this question, we need sufficient theory to begin to understand the gap between those adopters that abandon an innovation after its initial use and those adopters that make routine use of an innovation. The extant literature underscores the reasons why ASD methods are initially adopted (Tessem 2003; Bahli and Zeid 2005; Svensson and Höst 2005), however, there is a dearth of empirical investigation concerning the routine use of ASD in organizations despite calls for a greater understanding of ASD adoption beyond the initial stages (Abrahamsson et al. 2009). For instance, a prior empirical investigation suggests that the comparative advantage of an innovation over its predecessor is a major reason that an organization may adopt agile over other ISD methods (Rogers 2003; Senapathi and Srinivasan 2012). However, this comparative advantage does not adequately explain routine usage decisions in organizations that have decided to forego the use of ASD methods while achieving comparative benefits of agile

over other ISD methods. The literature lacks an understanding of the factors that separate those adopters that, despite experiencing the promised benefits of ASD methods, choose not to use ASD methods on a routine basis and those adopters that use ASD methods on a routine basis. One approach that allows us to examine this gap between adopters is the research on assimilation gaps. The research on assimilation gaps focuses on understanding the lack of congruence between the extent to which organizations commit to adopting an innovation versus how infused it is within the organization. One particular theoretical lens that we leverage to complement the research on assimilation gaps is the diffusion of innovation theory. As a theoretical lens, diffusion of innovation theory examines the extent that ASD represents an innovation in terms of a fundamental shift in the way that ISD has typically been done. In line with prior research, we use diffusion of innovation theory to provide some insight into the progression in which an innovation proliferates throughout an organization. With respect to ASD, the extant research using diffusion of innovations theory provides insights such as the motivation for ASD adoption in organizations, the adaptations that should accommodate ASD implementations, and the results of ASD use on project outcomes. However, there is little knowledge concerning the factors that affect ASD routinization intentions after initial use. This insight is critical as more organizations pour resources into making ASD their principal method of ISD. Additionally, the assumption that successful ASD use will lead to further ASD use has not held, as some organizations have abandoned ASD use after relatively successful ASD usage efforts (Sutherland 2014). Thus, we couple the research on diffusion of innovations theory with the research on assimilation gaps to study the

factors that heavily influence the routine ASD usage intentions in organizations. Key terms associated with the diffusion of ASD methods are defined in Appendix B1.

Prior research suggests that organizational factors play a pivotal role in the diffusion of an innovation within an organization (Chan and Thong 2009). In reference to ASD methods, both practitioner surveys and prior research highlight the influential roles of organizational culture and organizational structure on the assimilation of ASD methods. For instance, a recent survey suggests the impact of organizational culture as a critical factor in the ASD assimilation process, during both early and late stages (Ambler 2014). Additionally, agile's emphasis on social interactions and continual readiness towards changing requirements (Beck et al. 2001; Conboy 2009) has been shown to induce significant changes to more hierarchical organizational structures (Nerur 2005). Despite widespread success stories across differing organizations, in recent surveys, stakeholders with years of experience using ASD methods rated the "inability to change an organization's culture" as a key impediment to ASD assimilation (Ambler 2014; Version One 2014). Although many studies have recognized the compatibilities and incompatibilities between ASD methods and specific organizational culture types (Iivari and Iivari 2011), only a dearth of studies have examined this relationship empirically. We contend that organizational factors of culture and structure, which undergird the entire ASD assimilation process, play a significant role in the routinization of ASD methods. Therefore, we employ a final theoretical lens; that of organizational culture theory to examine the relationship between organizational factors of culture and structure, and their

influence on the ASD assimilation process. We are going to discuss these theories in greater depth in the next section.

In summary, this study seeks to close two significant knowledge gaps in the current body of ASD research by provide the following contributions: 1) providing an understanding of ASD assimilation beyond initial use, and 2) empirically investigating the relationship between organizational factors of culture and structure, and the routinization of ASD methods. These gaps motivate the following research question: *how do organizational culture and structure affect the routinization of ASD practices in organizations?* Drawing on prior research, we employ a theory building approach using the social process model as a framework to investigate ASD assimilation gaps. Our purpose is to provide clarity concerning the organizational factors that influence the routinization of ASD methods. To accomplish this, we compare and contrast the impact of multiple contextual factors on multiple implementation stages (Cooper and Zmud 1990). We expect that a careful examination of the contextual factors at play will yield helpful and new insights on ASD assimilation. Although our primary focus is to extend the literature by investigating the gap between the initial use and routinization of ASD methods, our study is exploratory and will report any findings on other assimilation gaps as well (Kwon and Zmud 1987).

The remainder of the article proceeds as follows. First, the theoretical foundations section will provide an overview of past research and relevant literature on the theories used to answer our research question. Second, the research methodology section will lay out the case study design strategy and the pertinent details concerning each company. Third, the

case analysis section will provide a cross-case analysis of ASD adoption scenarios at each site. Fourth, the discussion section will elaborate on the research findings by presenting a new framework and considerations for future research. The next section provides a brief over of the ASD assimilation literature, which exposes the knowledge gaps that this research intends to address.

THEORETICAL FOUNDATION

In this section, we give an overview of the research on assimilation theory, assimilation gaps, and organizational culture theory, which were used as the basis for this work.

Assimilation Theory

Innovation Assimilation

The extant literature extends diffusion of innovation theory (DOI) (Rogers 1962) from individual to organizational-level adoption (Rogers 2003) to study the diffusion of complex innovations in organizations (Kwon and Zmud 1987; Meyer and Goes 1988; Cooper and Zmud 1990; Gallivan 2001). We refer to the research on organizational-level assimilation of innovations as *assimilation theory* (Wang et al. 2012). The research model developed by Kwon and Zmud (1987) and further refined by Cooper and Zmud (1990) defines *six-stages of innovation assimilation* that describe technology implementations in organizations. *Innovation assimilation (IA)* is defined as the extent to which the use of an innovation diffuses across "the organizational projects or work processes and becomes routinized in the activities of those projects and processes" (Purvis et al. 2001). Table 3.1

presents the six-stages of the IA model, defining each stage of the process and delineating the adoptive use stages from the post-adoptive use stages. Each stage describes a differing level at which an innovation diffuses an adopting unit (Overhage and Schlauderer 2012). Extant literature relates *adoptive use* of an innovation to the initial three phases (initiation, adoption, adaptation) and *post-adoptive use* and implementation to the last three phases (acceptance, routinization, infusion) (Kwon and Zmud 1987). Like Roger's (1962) original DOI model, Kwon and Zmud's (1987) six-stage IA model has been adapted to study not only technology innovations, but also ideas and practices. ASD methods can be considered an ISD innovation (Chan and Thong 2009) because they represent a significant departure from previous methods for building software, as determined by the collective judgements of experts in the field (Meyers and Goes 1988; Beck et al. 2001). Moreover, the implications of studying ASD as an innovation and not just as a method potentially yield insights as to the broader impacts of ASD implementations across organizations. Thus, in line with prior research, we study ASD as an ISD innovation and the degree to which it is assimilated in an organization (Wang et al. 2012; Senapathi and Srinivasan 2012).

Table 3.1. Definitions of the Assimilation Stages (Cooper and Zmud; Senapathi and Srinivasan 2012)

Innovation Assimilation Stages

Adoptive
1. Initiation: need for change is recognized, a match is identified between an innovation and its application in the organization
2. Adoption: a decision is made to adopt an innovation
3. Adaptation: an adaptation to suit the contextual needs

Post-adoptive
4. Acceptance: use of the innovation
5. Routinization: an increase in the extent and intensity of use
6. Infusion: increased usage in a more comprehensive and integrated manner results in increased effectiveness of systems development

ASD assimilation

Because our focus is on the organization-level, we draw upon IA research to study the diffusion of ASD methods in organizations (Kwon and Zmud 1987). We define *ASD assimilation* as the extent to which the use of ASD methods diffuse across the organizational projects or work processes and becomes routinized in the activities of those projects and processes (Cooper and Zmud 1990; Wang et al. 2012). With respect to Kwon and Zmud's (1987) six-stage IA model, the early ASD literature focuses primarily on understanding the adoptive use stages (initiation, adoption, and adaption) of ASD assimilation (Tessem 2003; Bahli and Zeid 2005; Svensson and Höst 2005; Nerur et al. 2005), while later studies provide insights at the post-adoptive use stages (acceptance, routinization, infusion) (Chan and Thong 2009; Mangalaraj et al. 2009; Vijayasarathy and Turk 2012; Overhage and Schlauderer 2012). The studies that apply assimilation theories at the post-adoptive ASD use stages largely aim to understand the antecedents that affect ASD use (acceptance). These antecedents differ depending on the theoretical perspective and level of analysis employed by the researchers. Despite their differences, these studies broadly agree that: 1) the antecedents that drive an innovation across the adoptive stages differ from those that influence the post-adoptive stages, and 2) further research of the downstream phases of ASD assimilation is needed to understand how to sustain agile use long-term. Indeed, empirical research is particularly lacking beyond the acceptance stage.

Although few in number, two approaches to studying ASD assimilation beyond the acceptance stage include measuring the impact of sustained ASD use on ASD outcomes,

and studying the contextual factors that impact the downstream phases of ASD assimilation. In the former approach, Senapathi and Srinivasan (2012, 2014) create, refine, and test their Agile Usage Model using an input-process-output (IPO) framework to empirically examine the impact of ASD outcomes. They also expand the construct of ASD use to include both measures of the intensity (vertical) and extent (horizontal) of use in organizations. Their results suggest that the relative advantage of ASD over its predecessor, and the aid of an agile coach are among the most influential factors that affect ASD use. Additionally, their results suggest that post-adoptive ASD use has significant impacts on productivity, predictability, and software development quality.

In the latter approach, Wang et al. (2012) draw on the IA model to further define the later IA stages (acceptance, routinization, and infusion) in the context of an ASD implementation, and identify the extent that specific ASD techniques are used within those stages by ASD adopters. For instance, they examine the way ASD techniques such as stand-up meetings and refactoring are used at each of the post-adoptive use stages of the ASD assimilation process. Their study illustrates both the value and further need to understand the contextual factors of ASD implementation.

Here we employ the latter approach to address the need for a context-rich, empirical explanation of key factors that influence the penetration of ASD methods beyond the acceptance stage. Thus, we discuss limitations of past research and ways in which the research can be extended as follows. First, the Agile Usage Model developed by Senapathi, Srinivasan, and colleagues (Senapathi and Srinivasan 2012; Senapathi et al.

2013; Senapathi and Srinivasan 2014) is useful for explaining the impact of deep ASD usage on outcomes, but it does not explain how the transition is made from one post-adoptive use stage to the next at the organizational-level. Although prior research confirms the utility of the model, it also points out that the model does not specifically draw out contextual factors that impact ASD assimilation such as the role of decision making practices in the organizations, the management style, including recognition and reward structures, hierarchy and bureaucracy, organization size, and overall enterprise governance (Russo et al. 2013). Additionally, most of the antecedents to ASD use in the model were derived from the DOI model, which tests early adoptive behavior in individuals (Rogers 1962). Of these antecedents, only two - relative advantage and agile coach - were confirmed during the testing of the model (Senapathi and Srinivasan 2014). Therefore, the Agile Usage Model needs to be extended to include both a context-rich and empirically validated set of factors of post-adoptive ASD use (Wang et al. 2012; Russo et al. 2013).

Second, the six-stage IA model provides a structured mechanism to analyze ASD method use, "respecting the incremental nature of adoption as opposed to an overly simplistic binary perspective" (Gallivan 2001). However, while the use of stage models aids our understanding of how an innovation becomes embedded into workplace routines (Saga and Zmud 1994), little research has been conducted to identify the gaps between stages of ASD assimilation, known as assimilation gaps. *Assimilation gaps* can be defined as the difference between the patterns of cumulative assimilation events of an innovation across a population of adopters (Fichman and Kemerer 1999). With respect to the gaps between

the acceptance and routinization stages in the ASD assimilation process, there is a need for examination of the differences between those organizations that elect to routinize ASD methods from those that elect not to routinize after successful completion of the acceptance stage. In our study, we refer to this assimilation gap as the *acceptance-routinization gap,* where *ASD acceptance* refers to the introductory adoption and employment of ASD methods as a process innovation for one or more software development projects (Vijayasarathy and Turk 2012; Wang et al. 2012), and *ASD routinization* refers to the use of ASD methods as a normal activity in an organization (Senapathi and Srinivasan 2012). The IS field's understanding of this gap is unfortunately limited, yet knowledge concerning it is of particular consequence to practitioners looking to sustain ASD practices. The extant literature is missing an understanding of *how* the transition from the acceptance to routinization stages occurs and *why* some organizations elect to adopt ASD methods, not only initially, but also on a continual basis. As a result, there is a need for understanding the factors that affect the ASD routinization intentions after initial use (Wang et al. 2012).

Using the research on assimilation gaps, we can complement the IA model by studying the forces that influence the diffusion of ASD methods between assimilation stages. Indeed, one of the limitations of the stage models upon which the assimilation theories are based on is the neglect of details between each stage (Sabherwal and Robey 1995). As a result, "the ASD assimilation process illustrates the path of assimilation, but it cannot answer, at least by itself, questions such as 'how and why the assimilation of practice progress from one stage to another?'" (Wang et al. 2012). According to the research on

assimilation gaps, substantial gaps can present a misleading image of the diffusion process, which leads to inaccurate conclusions about the strength of the diffusion process that is being observed. Consequently, erroneous theoretical and practical inferences may be drawn based on false assumptions (Fichman and Kemerer 1999). For instance, using the IA model by itself, one might assume that an innovation that successfully diffuses through one assimilation stage will automatically advance to the next. However, regarding the acceptance-routinization gap, practitioner literature suggests that some organizations elect not to routinize ASD methods after successfully completing the acceptance stage (Denning 2012a, 2012b; Sutherland 2014).

Thus, this study combines the assimilation theories and the research on assimilation gaps to understand the process and the factors that influence the ASD assimilation process in organizations. The objective of this study is to provide a context-rich understanding of the organizational factors that influence the intent to routinize ASD methods after initial ASD use. To accomplish this, we leverage the strengths of the IA research --- to examine the concept of *use*, not in a sense of use vs. non-use, but rather the extent to which an innovation is used and how its use influences an organization's practices, structures, and organizational culture (Gallivan 2001) --- to study the acceptance-routinization gap.

Organizational Influences on ISD

An organizational-level perspective on ASD is important for understanding the ASD assimilation process because ASD projects take place in an environment that is broader than the project itself (PMBOK 2013). Therefore, we must take into consideration how ASD projects are carried out in alignment with an organization's mission, goals, and

objectives. Compared to traditional ISD methods, ASD methods require greater levels of communication, cooperation, and coordination from a cross-organizational team, which may have a greater impact on existing organizational structures and cultures than traditional ISD methods. Therefore, we review the literature to gain a better understanding of the relationship between organizational culture and structure.

Organizational culture theory

To investigate the relationship between OC and post-adoptive ASD use, we draw on the organizational culture (OC) literature to highlight the composition of OC, the different OC orientations, and the effect of OC on the use of ASD methods. Although organization culture's broad conceptualization makes it arguably connected to every organizational process, OC's complex, interrelated, and somewhat ambiguous set of factors make it impossible to create a comprehensive framework (Cameron and Quinn 2011). Consequently, researchers have had marginal success integrating and organizing elements of OC into widely used frameworks. One exception is Schein's (1985) three-layer model of basic assumptions, espoused values, and artifacts, which conceptualizes observable and unobservable layers of OC. Using Schein's (1985) definition of group culture, we define OC as "a pattern of shared basic assumptions learned by an [organization] as it solves its problems of external adaptation and internal integration, which has worked well enough to be considered valid and, therefore, to new members as the correct way to perceive, think, and feel in relation to those problems". His three-layer model suggests that aspects of OC can be tapped by focusing on manifesting elements that are theoretically and practically tied to the unobservable layers. Thus, the vast majority of

OC studies focus on values as constituents of OC (Hofstede 1980; Quinn and McGrath 1985).

Following prior ISD research (Iivari and Huisman 2007), we draw upon the Competing Values Model (CVM) to conceptualize OC as the manifestation of competing value systems. These culture types form from four quadrants, differentiated by the continuums of two dimensions, internal to external focus and change to stability as shown in Figure 3.1 (Iivari and Huisman 2007). Each of the four quadrants in the CVM represents a distinct OC, which we will draw upon to differentiate opposing OCs according to their core of values. We use Iivari and Huisman's (2007) categorizations of OC as shown in Table 3.2.

Table 3.2. CVM Categorizations
Group culture (change and internal focus) is primarily concerned with human relations and flexibility. Belonging, trust, and participation are its core values. Effectiveness criteria include the development of human potential and member commitment.
Developmental culture (change and external focus) is future-oriented, considering what might be. The effectiveness criteria emphasize growth, resource acquisition, creativity and adaptation to the external environment.
Hierarchical culture (stability and internal focus) is oriented toward security, order, and routinization. It emphasizes control, stability and efficiency through the following of regulations.
Rational culture (stability and external focus) is achievement-oriented, focusing on productivity, efficiency, and goal achievement.

In our research, we focus specifically on the interaction of ASD methods and hierarchical and developmental cultures for the following reasons: First, our primary proposition is that ASD values compete and conflict with those of a hierarchical culture. Thus, a hierarchical culture should have a negative influence on ASD routinization. Given that

the hierarchical and developmental cultures are polar opposites in the CVM framework, these culture types pose competing and conflicting demands on organizations (Iivari and Huisman 2007). Second, from a competing values standpoint, the emphasis of the developmental culture closely aligns with the tenets of the ASD philosophy as espoused in the Agile Manifesto. Thus, we chose a more parsimonious approach to test our hypothesis by focusing on the hierarchical and development cultures to discriminate between OCs based on their most salient differences. These differences manifest themselves in an organization's structure, which we will discuss in the next section.
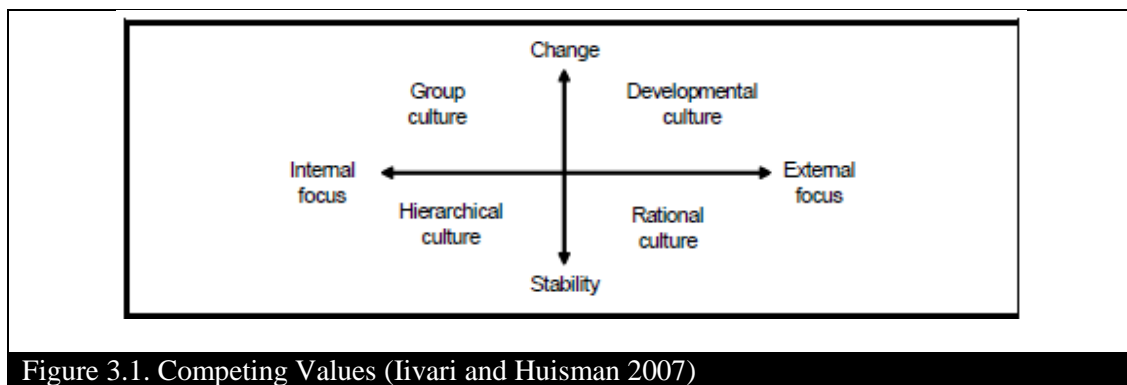
Figure 3.1. Competing Values (Iivari and Huisman 2007)

Organizational Structure

Organizational structure (OS) can be defined as "the logically consistent clustering of an organization's elements as it searches for harmony in its internal processes and consonance with its environment" (Mintzberg 1980). According the project management body of knowledge (PMBOK 2013), organizational structures range from functional to projectized, and can affect the availability of resources and influence how projects are conducted (PMBOK 2013). With respect to prior research, we postulate that an organization will produce an organizational structure that is reflective of its OC orientation (Martin 1992; Sackmann 1992). Second, we postulate that the relationships

between OS and ASD routinization will parallel the relationship between OC and ASD, as OS is a manifestation of OC. Thus, we match the OC orientations with their corresponding OS as follows: hierarchical OC to functional OS and developmental OC to projectized OS. The PMBOK (2013) defines these structures and their hybrids as shown in Table 3.3.

Table 3.3. Organizational Structures in this Study
Functional OS is a hierarchical organization where each employee has one clear superior, and staff are grouped by areas of specialization and managed by a person with expertise in that area.   • Strong matrix – an organizations have many of the characteristics of the projectized organization, and have a full-time project managers with considerable authority and full-time project administrative staff.
Projectized OS is an organization in which the project manager has full authority to assign priorities, apply resources, and direct the work of persons assigned to the project.   • Weak matrix – an organizations maintain many of the characteristics of a functional organization, and the role of the project manager is more of a coordinator or expediter

Given our primary proposition, we propose that a hierarchical culture (Quinn 1988) predicated on bureaucracy, should produce OSs to support the culture. On the other hand, we propose that a developmental culture (Quinn 1988) predicated on adhocracy, should produce OSs to support the culture. Prior research suggests that these bottom-up cultures are more compatible with ASD methods (Iivari and Iivari 2011), which support our

primary proposition. Table 3.4 summarizes that difference between the hierarchical and developmental cultures, and their resulting OSs based on the competing values model (Quinn and Cameron 2011).

Table 3.4. Hierarchical vs. Developmental in the Competing Values Model		
Organizational Culture	Hierarchy	Developmental
Organizational Strategy	Specified and predictable	Flexibility and adaptive
Enterprise Governance Structure	Bureaucratic	Adhocracy
Team Management Structure	Functional	Projectized
Control Mode	Formal	Informal
Organizational Focus	Internal	External
Stability vs. Change	Stability	Change

The Impact of Organizational Culture on ASD

In the IS literature, culture has been identified as an influencer of the use of information technology at the national, organizational, and group levels (Leidner and Kayworth 2006). In this study, we focus on culture at the organizational level and its interaction with post-adoptive ASD use, using the research on innovation theory (Rogers 1983; Kwon and Zmud 1987). Prior research has shown interest in the relationship between cultural values and the diffusion of IT (Kitchell 1995; Hoffman and Klepper 2000). Commonly, findings suggest that different cultural orientations affect the propensity of IT adoption. Researchers have also examined the impact of culture on ISD process improvement. For example, Dubé and colleagues studied the compatibility between organizational values and values subgroups (Dubé 1998, Dubé and Robey 1999) with results suggesting that high compatibility between the organizational values and a given process innovation, the more successful the implementation is likely to be.

More recently, the ISD researchers have studied the relationship between OC and the deployment of ISD methods (Iivari and Huisman 2007). Regarding ASD methods, most researchers have focused on the relationship between OC and ASD use (Robinson and Sharp 2005, Tolfo and Wazlawick 2008, and Strode et al. 2009), which is usually studied in the early adoptive stages or simply use versus non-use, with few exceptions (Iivari and Iivari 2011). However, despite calls for future empirical investigation on the influence of OC on post adoptive ASD (Chan and Thong 2009; Iivari and Iivari 2011; Vijayasarathy and Turk 2012) there still remains only a dearth of empirical examination on this relationship. Additionally, IA research on ASD suggests that OC plays a key role in the continual use of an innovation, however, no empirical studies were found that explore the relationship between OC and post-adoptive use (routinization and infusion) specifically. Therefore, we revisit the relationship between OC and ASD assimilation by studying various levels of adopters and non-adopters with the intention of theory building. We intend to investigate the influence of organizational factors (culture and structure) on the routinization of ASD methods. Our methodology section will describe our approach to studying the major gap between the acceptance and routinization stages in ASD assimilation, and the influential role of organizational factors of culture and structure. This research, which is exploratory in nature, will also note other contributing factors to the acceptance-routinization gap and other assimilation gap.

RESEARCH METHODOLOGY

This field study (Klein and Meyers 1999) employs an exploratory case study methodology for systematically investigating the ASD phenomenon in its real life context (Yin 2008). Given ASD assimilation's nascent stage of knowledge, we used an inductive, interpretive approach to study the adoption of ASD methods using semi-structured, face-to-face interviews; participant observation, and informal meetings to collect data from four separate organizations. This study emphasizes software development as a process involving the coordination of people with different values, expectations, and skillsets, each with his or her own frame of interpretation (Vidgen and Wang 2009). A multi-case study allows us to examine the results of different ASD implementations in various bounded systems. The cases were set up at four different sites to avoid selection bias and to show multiple perspectives on the issue of ASD assimilation. Our research design allows the use of replication logic, as the details of the cases are replicated and analyzed using the same theoretical lens.

Case site selection

Potential participant organizations were identified through software development community engagement, IT practitioner conference participation, and personal and professional networks. The lead researcher requested the participation of these organizations mostly through the face-to-face engagement with members that represented their respective organizations at community events. Preliminary emails were sent to potential participants who were either not available to meet face-to-face or no one on the

research team had direct access to. In our initial engagement, we asked about the organization's history with ASD methods and in most cases, requested permission to interview stakeholders that participated in ASD projects.

Case site selection criteria

Separate criteria were established to distinguish case sites in the introductory stages of ASD assimilation from those in the more advanced stages. Those organizations in the introductory phases had to first demonstrate their commitment to implementing an ASD methodology by at least three months after the our initial date of contact in order to fit into the data collection phase for this research. Second, organizations had to agree to provide the researcher with access to at least one member of each functional organization of the company represented in the cross-functional ASD team. For example, an ASD team that consists of 3 developers, 2 quality assurance testers, a scrum master, and a product owner had to agree to grant the researcher an interview with at least one member of each role (developer, tester, scrum master, and product owner).

For organizations in the latter stages of ASD assimilation, the following criteria had to be met. First, an organization had to show current use of ASD methods in its ISD processes. Second, the site had to show examples of projects that were completed using ASD methods. Third, key informants had to demonstrate knowledge of ASD practices. For instance, the researcher asked each informant about their length of experience and level of involvement with ASD methods. Each informant was asked to provide an example of a specific ASD method in which they had experience with. This was used to differentiate

between those that claimed to implement ASD methods, but in actuality only used ASD tools, or may have been mistaken concerning what ASD methods are. Our notion of what constituted "agile" included any software development method that ascribed to the basic tenets of the agile manifesto and included foundational elements of iterative development, self-organization, and process flexibility (Dissanayake et al. 2013). This included adaptions of some of the most widely used methods such as Scrum and XP.

In total, 20 different companies were identified and contacted with seven meeting our baseline criteria. Sites were removed if they lacked ASD implementation experience, were not currently committed to ASD methods, or planning to use ASD methods in the near future. In the end, four sites were carefully selected after consulting with key informants involved in ASD projects at each company. We strategically selected to interview ASD teams at different stages of the assimilation process to we compare and contrast the impact of multiple contextual factors on multiple implementation stages (Cooper and Zmud 1990). The next section provides a brief description of the four companies as follows as shown in Table 3.5.

Case site description

Table 3.5. Data Sources				
	Alpha	Beta	Dynamic	Century
Projected number of Interviews	5-10	17	17	7+

	Alpha	Beta	Dynamic	Century

(Table 3.5 continued)

Informants	Sample of Key Informants: Software Architect Software Dev Mgr. (3) Applications Dev Lead QA Director QA Manager	Development Team: Developers (5) QA Tester (4) Scrum Masters (2) Product Owners (2) Code Quality Specialists UI Designer Enterprise Architect Agile Coach	Census: Developers (9) QA tester (2) Project Managers (2) Creative Designer Interactive Designer Technical Developer Mgr CEO	Sample of Key Informants/ Development Team: Developers (2) VP of Operations (1) Project Manager (1) Director, Apps Development and Enterprise Apps (1) Director of Business Development (1) VP of Sales (1)
Avg. Agile Team Size	10	17	5-7	4-5
Organization Size	15,000+	15,000+	28	100+
Project Length	3 years	1 year	1-36 months	1-36 months
Development Scope	In-house	In-house	External Customer	External Customer
Industry	Healthcare	Healthcare	Software Development	Software Development
Agile Technique	Waterfall/Scrum Blend	Scrum	XP	Scrum

Alpha

Alpha is a large healthcare enterprise that is over 30 years old and offers healthcare services in the U.S.A. Alpha's IT department consists of over 125 staff members, who develop, deploy, and maintain a number of in-house applications, which are used by the company's healthcare professionals. Alpha contains a weak matrix OS, and is considered to be a highly government regulated and bureaucratic organization by the interviewees.

The company had had a previous ASD implementation that was widely considered a "colossal failure".

Beta

Beta is a large healthcare enterprise that is 75 years old and offers healthcare services in the U.S.A. Beta's IT team has over 300 staff members. The company contains a weak matrix OS, where project managers fulfill specific project roles, but functional managers maintain authority over ISD projects. The company is considered to be a highly government regulated and bureaucratic organization by the interviewees. The company had committed to employing Scrum on a mission critical project after a previously failed attempt using waterfall.

Century

Century Technology is a medium sized IT services company in the U.S.A. that is under 20 years old. Century's IT team has over total 100 staff members that manage its consulting, solutions, and support operations. The company is considered to be a team oriented and people-centric organization by the interviewees. Century contains a strong matrix OS, where project managers control most aspects of ISD projects. The company has been using ASD methods for 2 years with great success, and is now attempting to diffuse ASD principles in its other workflows.

Dynamic

Dynamic Enterprises is approximately 11 years old and offers creative services in the U.S.A. Dynamic's team has 27 staff members that specialize in custom software and design services. The company contains a projectized OS and is considered to be "dynamic" and "forward-looking" by its employees. The company has been using ASD methods for over 5 years with great success.

The sites and the researcher

The lead researcher accumulated over 1000 hours of field research during this study. First, the lead researcher met and cultivated relationships with the software development community by attending .NET user group meeting on a monthly basis over a four-year period (2011-2015). The lead research was invited to give a presentation on the state of ASD research, and solicit feedback from the practitioner community, including current challenges in routinizing ASD methods. The lead researcher was later able to schedule individual attendees for face-to-face interviews.

Second, the lead researcher previously interned at Alpha and was employed at Beta (unrelated to software development) during the time of this study. Therefore, the researcher was able to gain firsthand experience and knowledge about each organization's company culture, communication structures, reward and incentive structures, governance structures, and disposition towards ASD methods. In addition, these experiences allowed the research to observe the day-to-day operations of each organization, which provided insights beyond what could have been understood from the

outside. This translated over to the interview process, where interviewees were familiar with the interviewer and saw him as a fellow member instead of an outsider. Thus, interviewees tended to use specific names instead of position titles and appeared to be less guarded. For the researcher, he was able to ask questions that are more specific and gain insights concerning specific projects (by code name) during the interview. In total, the lead researcher spent 8 months at Alpha and 10 months at Beta. During the time that the research spent at both sites, he observed each organization's software development practice and engaged in note taking, recording interviews (transcribing later), and examining artifacts, records, and ceremonies.

Third, the lead researcher's involvements with the ISD community user groups led to opportunities to visit the sites of Century and Dynamic. The lead researcher toured each site multiple times, met with members across different departments/roles, and interviewed members on-site during normal workdays. Follow up interviews were conducted as needed.

Interviewees were selected from each organization of those who either participated in an ASD implementation or were highly affected by the efforts. Ethical guidelines were followed. In depth interviews were taped and transcribed after being conducted at each site. The duration of the interviews ranged from 10 to 50 minutes for individuals and up to 2 hours for groups as shown in Table 3.6. Additional documents and artifacts were collected as they were made available to the researcher. Appendix B2 shows a sample of the open-ended and semi-structured questions used to guide each interview. A total of 44

individual interviews and six group interviews were conducted across all four sites.

Altogether, the data collection efforts took place from May 2014 to December 2014. The

next section describes the framework that was used to analyze ASD implementations at

each of the four case sites.

Table 3.6. Interview Details		
Alpha's Respondents	Number of formal interviews	Duration
Alpha.Architect.1	1	i
Alpha.ISDMgr1	1	i/g
Alpha.ISDMgr2	1	36:33
Alpha.ISDMgr3	1	g
Alpha.ISDMgr4	1	1:02:11 i/g
Alpha.QAMgr1	1	10:02
Alpha.QAMgr2	1	i/g
Alpha.ISDMgr5	1	g
Alpha.ISDMgr6	1	1:16:20
Beta's Respondents	Number of formal interviews	Duration
Beta.Dev1	1	22:49
Beta.ITMgr1	1	19:19
Beta.Dev2	1	25:35
Beta.Dev3	1	31:24
Beta.PO1	1	21:47
Beta.PO2	1	Informal
Beta.Dev4	1	17:44
Beta.ITMgr2	1 interview; also 2 informal meetings; several interactions	25:25
Beta.PM	1	24:37
Beta.Dev5	1	16:31
Beta.Dev6	1	21:16
Beta.Dev7	1	27:35
Beta.QA1	1	28:22
Beta.QA2	1	21:47
Beta.QA3	1	20:44
Beta.Dev8	1	50:03
Beta.Coach1	2 interviews	Informal
Beta.Coach2	None; 1 informal meeting	Informal
Century's Respondents	Number of formal interviews	Duration
Century.Dev1	1	22:10
Century.Dev2	1	21:28

(Table 3.6 continued)

Century.Mgr1	1	22:07
Century.PM	1	25:14
Century.ISDMgr	1	42:59
Century.Mgr2	1	17:15
Century.Mgr3	1	30:31
Dynamic's Respondents	Number of formal interviews	Duration
Dynamic.Dev1	1	22:37
Dynamic.Dev2	1	19:53
Dynamic.Dev3	1	25:01
Dynamic.Dev4	1	18:19
Dynamic.Dev5	1	26:05
Dynamic.Dev6	1	29:06
Dynamic.Dev7	1	26:41
Dynamic.Dev8	1	13:54
Dynamic.Dev9	1	14:54
Dynamic.QA1	1	26:40
Dynamic.QA2	1	23:14
Dynamic.PM1	1	
Dynamic.PM2	1	
Dynamic.Des1	1	17:07
Dynamic.Des2	1	17:00
Dynamic.ISDMgr1	1	32:54
Dynamic.ISDMgr2	1	
Dynamic.Sales	1	

Protocol development and data collection

Following procedures outlined by Yin (1994), a framework inspired by Newman and

Robey (1992) was developed to guide our data collection efforts as shown in Figure 3.2.

The framework identifies *five* factors that were used to investigate the interaction

between corporate culture and the assimilation of a process innovation: History, context,

ASD acceptance, outcomes, and ASD routinization intentions (as described in Appendix

B3).
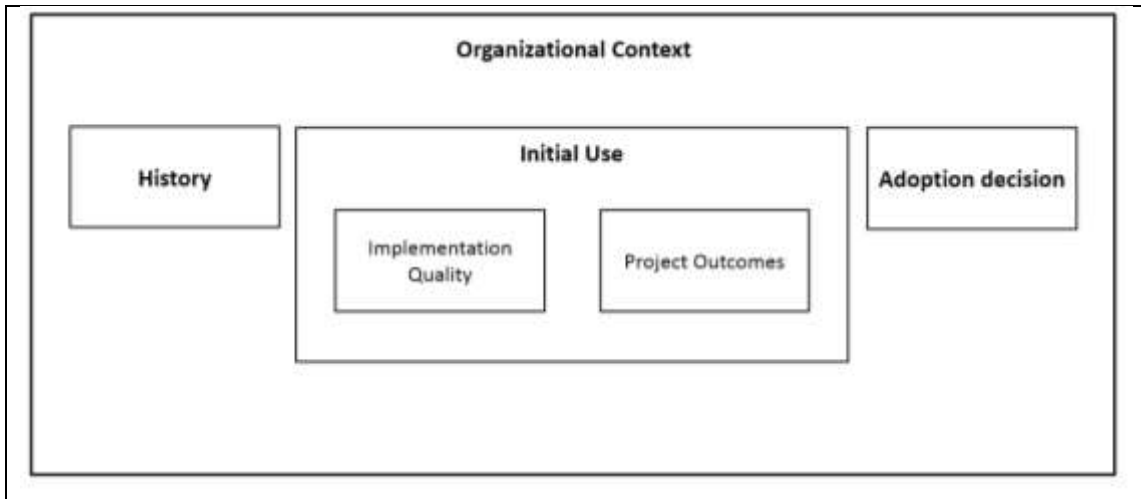
Figure 3.2. Modified Routinization of Agile Methodologies – Theoretical Model (Newman and Robey 1992)

CASE ANALYSIS

Theoretical Model (cross-case analysis)

History

Table 3.7. History				
Antecedent Conditions to Adoption Decisions	Alpha	Beta	Century	Dynamic
Popularity of agile methodologies			✓	✓
Failure or shortcoming of existing methodologies	✓	✓	✓	
Promise of Process improvement	✓	✓	✓	✓

In our analysis, many factors elucidated the ASD alternative to existing ISD methods as shown in Table 3.7. First, the sheer rise of agile's popularity and its use within the ISD industry made it a conspicuous alternative to companies using other methods, particularly waterfall. In our study, stakeholders including developers, IT analysts, project managers, and senior managers in every company were aware of the burgeoning popularity of ASD methods. For companies that compete on the cutting edge, agile represented a forward

way of thinking about ISD, as in the case of Dynamic. When asked why the company adopted ASD methods, Dynamic.Dev1 explained that,

> (Agile) kind of came up in the industry as a better way to do things and I think we try to keep an ear to the current situation our industry and so naturally, we could logically conceive that this made sense in a certain way.

Second, the awareness of ASD methods combined with the failure or shortcomings of other ISD methods, as in the case of Beta, further emphasized the ASD alternative, especially when an agile champion is present within an organization. For Beta, the failure of a mission critical project using waterfall led to senior management reconsidering the ASD alternative. Beta.ITMgr1 commented,

> (Agile) sort of hit mainstream simply because the project that we were working on last year completely failed. There were millions of dollars that were at stake, so we needed a change in process and so far this thing has been able to provide the quantified results in a short period of time for all the stakeholders to have visibility to.

Conversely, the ability to adjust to changing scope requirements influenced Century's initial use of ASD methods as the company struggled to meet customer expectations. Whether a single critical incident or a series of incidents, events that link an existing method to highly negative business impacts often lead to the consideration of alternatives (Vijayasarathy and Turk 2012).

Last, the promise of process improvement, even with the success of an existing method, influenced initial adoption decisions. For example, Alpha's desire to improve relationships between key stakeholders during ISD led Alpha to consider ASD methods. Alpha's IT and business department had a long history of distrust and communication

issues during ISD projects. Thus, as Alpha.ISDMgr6 explained, implementing ASD methods was an attempt to "improve interaction between the build team and the business." In line with both the academic and practitioner literature, all of the companies in this study adopted ASD methods to gain potential process improvements. The examination of each company's history provided both insights about the events that led to its consideration of ASD methods as well as a backdrop for understanding the specific corporate context in which the initial deliberations takes place.

Context

After a decision is made to adopt ASD methods, agile is then implemented into a particular corporate context. In this study, the interaction between the ASD method and the existing corporate context impacted the way in which the method was employed. In turn, the characteristics of this interaction not only affected the efficiency and effectiveness of the ASD implementation, but also the interpretations of the method's resulting project outcomes (Leidner and Kayworth 2006). In the next section, we describe the impacts of distinct types of OCs on the use of ASD methods and their subsequent adoption decisions. First, we were able to decipher each organization's culture by assessing its current activities and past history. Second, we differentiated opposing OCs by using the competing values model (Cameron and Quinn 2011). Third, we triangulated our characterization of each company using 10 aggregated dimensions for identifying and describing an organization's primary cultural characteristics (Larson and Gray 2010). This process, which is by no means exhaustive, helped us to tease out distinguishable characteristics of OC for the sake of comparison.

Hierarchical culture: Prior research has emphasized the importance of cultural compatibility in relationship to a given IS effort (Dubé 1998). That withstanding, the compatibility and relative fit of a top-down, hierarchy culture to a bottom-up method like agile, has been scrutinized (Boehm and Turner 2005; Nerur and Balijepally 2007). In this study, Alpha and Beta's hierarchical culture (Quinn 1988) predicated on bureaucracy, produced OSs to support the culture. Both companies were arranged in functional departments. These departmental "siloes" (Beta.PM1, Beta.PO2, Beta.Dev8, Beta.Dev1) supported each company's hierarchical reporting structures, but conflicted with ASD values as follows.

First, Beta struggled to operate in a collaborative fashion across its functional departments as work activities were organized around individual and not team efforts. Although the functional arrangement housed a network of professionals with specialized skills, the communication and distrust issues across departments hampered both organizations' ability to coordinate work activities effectively. Overall, members of each department adopted an "us versus them" mindset with respect to other departments. Beta.ITMgr1 explained his role in mediating disagreements between IT and business with,

> There was fighting. Constantly back and forth. My role is at the end of the day a family counselor in EA, always trying to work with the business and IT to work it out and to find that line in the sand that everybody can agree upon.

Second, both companies employed a high degree of organizational controls to both regulate member behaviors and govern practices. Beta governed its IT practices with a high degree of direct supervision, rules, policies, and rigid procedures. The company was

required to adhere to strict industry guidelines regarding the storing and transference of client information. Before ISD work began, all requirements were recorded in large requirements documents, which contained a fully defined scope of the project. Although the requirements document assisted developers in producing predictable results, and providing a safeguard for out scope requests, the document also facilitated autonomous, as opposed to, collaborative working patterns. Alpha.ISDMgr6 commented,

> It is like, the business has to control scope, (and) manage budget. I think that is why it sounds so outlandish when you tell somebody who has that much experience and has done things a certain way that everything is going to be good.

Similarly, Beta.PM commented,

> We are a very top heavy culture. We are very top heavy. We have a lot of managers. We have a lot of senior executive/executive folks that want to be in the weeds of decisions in different projects that we have going on and it may not be the best use of their time. This is just a historical issue that we have here is we need them to delegate authority and that doesn't always happen.

Third, in line with the competing values model, both hierarchical cultures placed an emphasis on stability over change. Key informants at Alpha and Beta described their prevailing organizational mindsets as risk averse and resistant to change. When asked to describe Beta's OC, Beta.Dev1 commented,

> We are a little bit more conservative. We are an older company, so there is some of that "this is the way we have always done it". We are probably a little slow to change, and slower to respond to things. We don't really trail blaze a lot.

> It has even been told to me that on many occasions that we do not like to be first with anything. We are just a little bit more risk averse and naturally so. We are a (healthcare services) company. The culture kind of supports that kind of mentality. If you are out there on the fringe and trying to be upon the bleeding edge, this culture will kind of reign you in a little bit.

Similarly, Alpha's IT Management was fixed on the main tenets of the waterfall process such as comprehensive documentation, contract negotiation, and following a plan. Alpha's attempt to introduce a new method in an environment that was built on stability proved to be a massive challenge. Alpha.Architect1 elaborated on the difficulty of using ASD methods routinely at Alpha with,

> I think it is just culture. The culture here is just so strong. Again, we have been doing things a certain way for so long. The mindset of software development as far as decision makers go, is old school. It can change, but [the push for change] has to be consistent. You have to keep hitting it, keep hitting it, keep hitting it… you can't hit , get a win, and then stop. If you don't [keep hitting it] and nobody is pushing it then it is never going to change. It is that strong.

As a result, neither company was able to introduce ASD methods into their existing hierarchical cultures as is. Beta was able to overcome its cultural challenges by creating an different subculture by forming a co-located and dedicated ASD team (Larson and Gray 2010). For Alpha, its inability to provide an environment that facilitated ASD workflows, to provide training to the members of its cross-functional team, or to address communication issues across departments exposed the ASD team to dysfunctions within the culture, which deeply affected the team's ability to execute Scrum effectively. Table 3.8 compares the hierarchical and developmental organizational structures that the opposing cultures gave rise to, based on the prior research of Russo et al. (2013), Iivari and Huisman (2007), and Gallivan (2001).

Table 3.8. A Comparison of the Hierarchical and Developmental OCs		
Organizational Culture	Hierarchy	Developmental
Enterprise Governance Structure	Bureaucratic	Adhocracy
Organizational Focus	Internal	External
Stability vs. Change	Stability	Change
Team Management Structure	Functional	Projectized
Decision Making Structure	Top-down	Bottom-up

(Table 3.8 continued)

Team Work Structure	Siloed	Collaborative
Recognition Structure	Individual specializations	Cross-functionality
Reward Structure	Individual performance	Team Success
Control Mode	Formal	Informal

Developmental culture: Century Technology and Dynamic represent a mostly bottom-up, developmental OCs (Quinn 1988), which prior research suggests are more compatible with ASD methods (Iivari and Iivari 2011). Unlike hierarchical cultures, developmental cultures are predicated on adhocracy, producing OSs to support the culture. Dynamic is arranged in a projectized OS. Although Century is arranged in a strong matrix structure because of its additional IT support services, its ISD services infrastructure resembles and functions similar to a projectized structure with dedicated project teams. Both OCs support projectized reporting structures, and facilitate ASD values as follows.

First, both Century and Dynamic were structured in a manner that facilitated collaborative work as most work activities were arranged in projects where teamwork was encouraged. Even the reporting structures and office layout promote collaborative work. Project teams were autonomous and received little supervision from management. Normally, ISD projects were completed in teams unless the size of the project was small enough for an individual to complete it with minimal effort. Overall, members at both companies adopted team oriented approaches easily, which facilitated cross-functional work. Dynamic.Dev5 commented,

> People that are able to work on a team, and are self-motivated tend to collaborate themselves in these type of groups. When they are focused and have a task at hand, things just get done.

Compared to someone who is very intelligent and knows what they are doing but doesn't mixed well with others. I think the culture we have here really fits agile well. Very highly collaborative.

Second, Century and Dynamic have developmental cultures employed fewer formal controls. Both companies primarily use formal procedures to document the workflows of their perspective ASD method in order to make their procedures easier to follow for new members. However, neither company was forced to conform to many industry imposed rules. In fact, Century's ISD managers saw the company's lack of industry restrictions as a key factor in its freedom in selecting an ISD method. As a software vendor, Century's decision to try ASD methods after years of using waterfall methods was made based on customer preference rather than externally imposed rules. When asked whether selling a new method to management was difficult, Century.ISDMgr described the company's autonomy as follows,

One thing that I will say that's good about Century is (that) we are a very autonomous company and we do have a lot of freedom on the projects for us internally. The (internal) sell (of ASD) isn't that bad, but figuring out how to make that work with a consulting environment where we have to sell it to our clients, that's the harder challenge.

In addition, neither company employed direct supervision, heavy rules, or rigid procedures to its ASD teams. Management concerned itself more with supplying the teams with the necessary tools for completing projects than regulating the individual behaviors. Both companies utilized project managers to provide oversight of the project, but Century also included a team lead in the form of a Scrum Master. Century's Scrum Master focused on the execution of the method, while the PM coordinated communication between the developers and the clients. Dynamic, which used a modified

version of XP, used the project manager to fulfill both responsibilities. However, the Dynamic's developers are given a high degree of autonomy to complete projects. Dynamic.QA1 explained,

> We are a bunch of young, 20 somethings sitting in here writing code. They trust that we are sitting here not on Facebook all day and then going home. The reason they trust us is because they should because at the end of the day we are pumping out code and we are pumping out good products and they recognize that. Everybody here comes in. They hold one another accountable.

Century and Dynamic contained sales personnel, which sold the client on the company's ability to perform the job. The sales personnel also provided client education concerning the ASD process including the roles, responsibilities, and each company's client expectations. After entering into a contract with a client, development began immediately instead waiting until all project requirements were gathered and documented. The ISD teams only focused on documenting the project specific details that it gathered from the client every two weeks, and any addition details that arose within each iteration. Both companies worked in two-week iterations, which facilitated collaborative working patterns.

Third, Century and Dynamic emphasized change over stability. As ISD service providers, both companies emphasized the importance of adapting to change in relation to their survival in the IT industry. Century.ISDMgr commented about the company's openness to change as follows,

> I think since we're predominantly IT, we do have a good bit of openness when it comes to change because change naturally happens with our career.

Century.Mgr3 added,

I would say that Century is a little bit more open to change. Primarily because we have a little bit younger demographic because of the industry that we are in. Most our average ages is middle 30s. The company has only been around for 17 years, so it is not that we have these age old tried and true --- we have always done it this way and that's the way we are going to do it. We haven't been doing something for 90 years and that is just the way we do it. So we are pretty about good about change and there has been a tremendous amount of change in the last year here, with people, with processes, and with technology. We have changed a lot. Sometimes it is not always easy and not everybody embraces that change as quickly as others, but I would say...we are not bad at it.

In the same fashion, Dynamic's Dynmaic.Des1 related the company's openness to change to its company culture as follows,

The company culture here is different than any other place that I have worked. It's flexible, open to change, which is really nice. A lot of company will be stuck in their ways and we have done it this way for so long, we are not changing it. Whereas here, if there is something better and we can all agree that it is better then why not go for it.

Similarly, Dynamic is not only open to change, but also innovative and risk taking. Whereas Century offers ISD services among other IT services, Dynamic solely provides ISD and complimentary design services to its clients. Thus, Dynamic focuses on being a cutting edge, industry leader in custom software development. Thus, adapting to change is not just a way to adjust to changing customer requirements, but also a part of the company's entrepreneurial strategy. Dynamic.Dev5 commented,

The culture is semi-competitive, which is great. We are always finding things to solve or talking about how to do things. I think those type of people work really well for agile.

As a result, Century and Dynamic were able to introduce ASD methods into their existing developmental cultures without undergoing major structural changes or insulating their

ASD teams. Thus, ASD methods proved to be a good fit for both company's existing culture.

ASD Acceptance (Initial Use)

In our analysis, the degree of discipline in which a method is employed impacted its implementation quality. As the extant literature suggests, process innovations that are implemented without proper stakeholder training, process tailoring, and knowledge management devices often result in deviations from best practices (Overeem 2014). These deviations were largest when method knowledge was lowest.

Alpha: Lack of methodology knowledge and facilitating subculture: Alpha's knowledge and experience with ASD methods was very small. During its initial ASD implementation, the company's ASD knowledge was contained in a single developer who although he had some experience with agile, did not have any experience leading an ASD implementation. In addition, the company failed to ensure an effective way to transfer knowledge from its experienced developer to the rest of the team or provide its pertinent stakeholders with method knowledge through outside training. Not only were developers subject to learning on the fly, but knowledge concerning the non-developer team roles such as the product owner role was absent altogether. This resulted in numerous deviations from best practices, which hurt the team's ability to execute Scrum. Alpha.ISDMgr1 commented,

> There was a leading best practice that we didn't necessarily stick to. One of the really intrugal parts of Scrum, which was changing or adding stuff to the scope during the run of an iteration. That's a tenet. You can't break that and we broke it…a lot.

In addition, Alpha's ASD project was deeply affected by the history and dysfunctions within its OC. Alpha's fractured relationship between IT and business along with its siloed OS stymied communication between departments. These communication issues manifested during the project as the team was not insulated from the overall culture. Ultimately, an empowered, yet underinvested business manager drove scope changes into perpetuity until as the project continued to accumulate more and more technical debt. This eventually led to a top-down decision, when Alpha's CIO commanded the IT staff to "stop talking to the business". This critical incident marked the end of Alpha's agile-like deployment and its return to back to waterfall.

Ironically, the project was deemed a success in terms of fulfilling the businesses need to replace a pen and paper system. However, not only was the project delivered grossly over budget, beyond scope, and behind schedule, but also virtually all the key informants interviewed in this study deemed the project a huge failure. The implementation quality and not the end result wearied the project team and soured many in the organization on ASD methods. When asked how the low implementation quality affected him, Alpha.ISDMgr5 replied,

> I quit! I am a rehire. I rehired on a different team. It was affecting my personal life! My wife told me that I needed to quit. That's how mad I would come home.

Beta: Methodology knowledge and facilitating subculture: Beta, who had similar communication issues as Alpha, was able to overcome its challenges by providing stakeholder education from agile coaches, and gaining the buy in of team members. By

comparison, Beta, which is similar to Alpha in terms of OC, organization size, and industry, hired a pair of agile coaches to train its ISD team in Scrum (agile). The pre-project training and the assistance of the agile coaches during the early phases of the ISD aided the team in understanding the differences between ASD and previous methods, and understanding their new roles, responsibilities, and expectations. Prior to the start of the project, the majority of Beta's ISD team knew "very little about agile" (Beta.Dev4; Beta.PO1).

By gaining a knowledge of Scrum through formal ASD training, Beta's ASD team was able to execute the method with discipline. Deviations, though rare, were made consciously and by the team's consensus. Having gained a methodological understanding from the agile coaches, the team tailored the ASD process and executed Scrum in a bureaucratic, highly controlled environment. Not only was the team able to form its own culture according to ASD values, but it was also able to challenge long-standing mindsets in the OC to the benefit of the project. The agile coaches were fundamental in that regard. As Beta.ITMgr1 put it,

> Very early on, one of the most powerful benefits of having the coaches in the room was that the coaches help us change our culture. I would say if there's one thing that we do differently on this enrollment team, than we do on any other project is that we have a different culture in this room. The ability to co-locate us and give us coaches that are dedicated helped us to kind of change our culture. And we wouldn't have done that on our own if it wasn't for the two coaches that came on site.

In the end, the team delivered a new system that either met or surpassed the original specifications of time, scope, and cost. The project was deemed as a huge success by virtually all the key stakeholders interviewed in this study, both formally and informally.

Unlike Alpha, Beta's implementation quality increased the team's buy in of ASD methods even though some employees not directly involved with the project remained skeptical of ASD methods.

Century and Dynamic: Facilitating subculture: By comparison, both Century and Dynamic saw major success in their initial implementations of ASD methods. We observed their respective OCs as facilitators of ASD values. Century's team oriented and people-centric culture meshed well with ASD values of collaboration and team autonomy. Led by experienced developers, Century's first ASD project was deemed highly successful in terms of time and scope, despite it being slightly over budget. Overall, the results from using ASD methods were consistent and significant improvements over waterfall.

Similarly, Dynamic's innovative and entrepreneurial culture fit well with ASD values of adaptation and constant improvement. The company's top management encourages its employees to stay on the cutting edge of ISD innovation. For Dynamic, ASD methods represent the most forward thinking and sensible way of producing high quality software. Unlike the other companies in this study, ASD methods were Dynamic's original ISD team method. For its first few years, the company had very few employees. By the time Dynamic grew enough to have an ISD team, the company's top managers had already bought into ASD principles. Thus, the company took the Agile Manifesto and created its own flavor of ASD to suit its working needs.

The company determined by consensus which ASD ceremonies to adopt and how often to employ them. In the end, Dynamic's flavor of ASD resulted in a string of ISD projects that were deemed successful, both in time, scope, and budget as well as in their implementation quality.

Outcome

We observed various outcomes from our four case sites as shown in Table 3.9. First, all of the companies in this study were able to deliver working software to their customers. Second, Beta, Century, and Dynamic added a new method to their ISD toolkit after initial use, yet only two of those three adopted ASD as a new method for future use. Third, each organization added to its history concerning the use of ASD methods within its organization. For Beta, Century, and Dynamic, they added mostly positive experiences concerning the success of ASD methods at their respective organizations. These initial projects concluded with post-mortem reports and lessons learned that added to the organization's knowledge base. Of these, only Alpha had a mostly negative experience with its initial use of "agile".

Table 3.9. Agile Implementation Results		
	Implementation Quality	Project Success Measures* Time-Scope-Budget
Alpha	Low	Significantly over [all]
Beta	Very High	Significantly under [all]
Century	Moderate	On-time, within scope, over budget
Dynamic	Moderate-High	Met project objectives
* = All projects were accepted by the end user		

Routinization Decision

ASD routinization is the use of ASD methods as a normal activity in an organization (Senapathi and Srinivasan 2012). In this study, we follow the social processes of four companies during ASD implementations that resulted in the intent to continue or discontinue using ASD methods. Both Century and Dynamic expressed their intention to make ASD methods their standard operating procedure for ISD, while Alpha abandoned ASD methods after an initial project failure and Beta suspended its use of ASD methods after a major project success. Thus, contrary to conceptualizations in the extant literature, we observed that neither a relatively successful end product nor a high quality implementation led to the routinization of ASD methods across all organizations (Khalifa and Verner 2000; Green and Hevner 1999). Instead, the social norms of an organization's culture and the perceived consequences of continual use impacted individual perceptions and their subsequent adoption or rejection of the method.

DISCUSSION

In reflecting on the results of our study, two themes emerged revolving around culture match and assimilation gaps.

Culture Match

The results of our study suggest that an organization culture orientation strongly impacts the routinization of ASD methods, which aligns with our theoretical model in Figure 3.3. Organizations such as Century and Dynamic contained OCs that aligned with ASD

values and facilitated ASD methods. Consequently, successful ASD implementations led to further use of the method in developmental cultures. On the contrary, organizations such as Alpha and Beta that contained OCs that conflicted with ASD values, and thus, failed to routinize ASD methods. Beta's ASD team was able to overcome the presiding OC temporarily by insulating the team, but the detractors soon raged against the continued use of the me3hod because of the changes it would require to their working norms. Thus, a strong culture match is arguably the single best determining factor of ASD routinization in this study as shown in Figure 3.3 and Table 3.10.
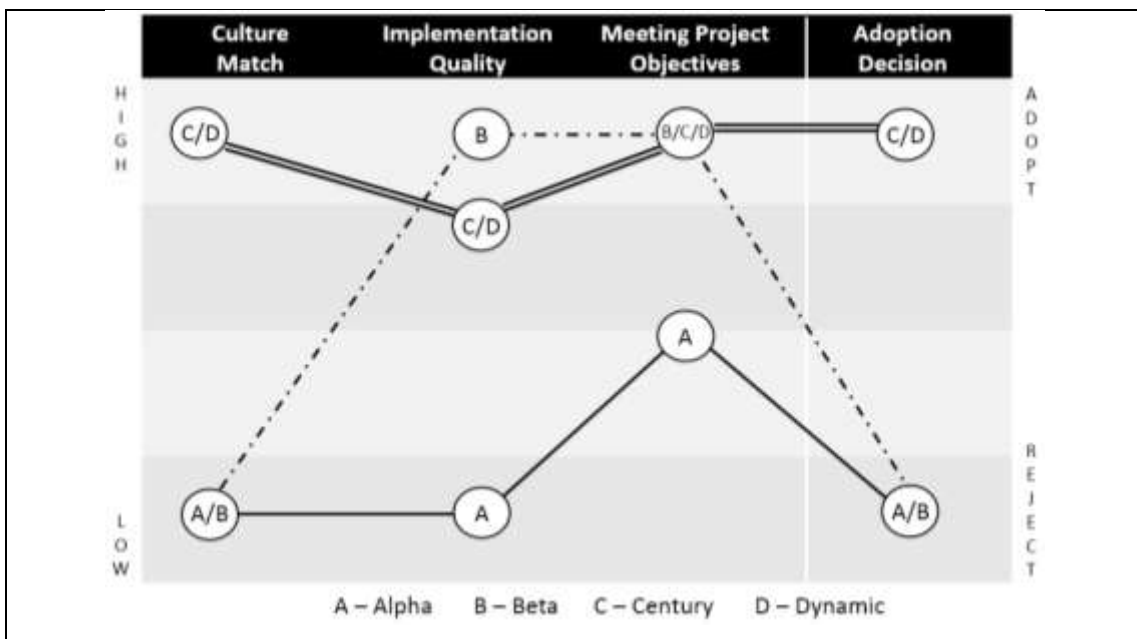
Figure 3.3. Agile Implementations on Adoption Decisions

	Culture Match	Implementation Quality	Meeting Project Objectives	Routinization Decision
Alpha	Low	Low	Medium	Reject
Beta	Low	High	High	Reject
Century	High	Medium-High	High	Accept
Dynamic	High	Medium-High	High	Accept

Table 3.10. Agile Implementations on Adoption Decisions

Assimilation Gaps

Although our primary focus is the acceptance-routinization gap, we identify and compare

ASD assimilation gaps across all four assimilators (as shown in Figure 3.3). Additional

assimilation gaps are discussed in Appendix B5. In investigating the reasons for these

gaps in ASD assimilation, certain factors were identified that either facilitated or impeded

the advancement of the diffusion process in an organization.

Acceptance to routinization gap – Lack of a facilitation OC (dominant)

As noted earlier, Beta's successful implementation of ASD methods did not lead to the

wholesale adoption of ASD methods by the company. Primarily, the company's hesitance

to routinize ASD methods hinged on its refusal to alter its organization structure to

accommodate the method. Beta.Dev4 explained,

> There are few people in the organization who are opening their minds to concepts
> of dedicated co-located teams focused on work, which is different than what has
> been our standard model for executing projects.

Ironically, information gathered during a company meeting revealed that a sister

company, Beta 2, located in a different region made the decision to adopt ASD methods

as the company's primarily ISD method. This move led to the alteration of the company's

OS from and a weak matrix to a strong matrix structure. In fact, project managers from

Beta visited Beta 2 in order to better understand their transition. However, the researcher

was only able to gather informal interview data and one artifact, during the presentation

on Beta 2's transformation. Thus, future research is warranted to gain a better

understanding of the transition process of Beta 2. Nevertheless, as we compare Beta to

Century, who is arranged in a strong matrix culture, we can observe the following. First,

according to the competing values model Century's culture has more of a developmental (major) and rational (minor) cultural characterization, in contrast to Beta's hierarchical culture. Second, after Century's initial ASD project success, the company made light alterations, instead of major changes to its organizational reporting structures. Century altered its project work structures and team reporting structures by assigning developers to dedicated teams that reported to a single team leader, and placing each agile team under the overall sight of a project manager. Thus, biggest difference between the two companies was that Century's OC and structure were primarily arranged to support project team efforts whereas Beta's was primarily arranged to support functional, departmental efforts. Thus, Beta's refusal to transition from a hierarchical to a more developmental culture and in turn, from a weak matrix to at least a strong matrix structure, stymied the diffusion of ASD methods; highlighting the acceptance-to-routinization gap as shown in Table 3.11.

Table 3.11. Agile Assimilation Gap Assessment				
	Alpha	Beta	Dynamic	Century
Agile Assimilation:				
Adopt Status	Agile rejecter	Uses sparingly	Full aadopter	Full aadopter
Project Team Status	Adoption	Acceptance	Routinization - Infusion	Infusion
Gap	Adoption – [X] adaption	Adaption – [X] Acceptance	Routinization – [X] Infusion	None
Inhibitors	Poor application and execution of the method; Lack of experience with methodology; Lack of training	Perception of outcomes: Impacts to organizational chart, reporting structure, and job roles; attribution of success to outside factors	Lack of formalization and mandatoriness	None

(Table 3.11 continued)

	Alpha	Beta	Dynamic	Century
Enablers	Stakeholder methodology knowledge	Training Facilitating subculture Internal customer buy in	Facilitating org. culture Customer buy in	Training Facilitating org. culture Customer buy in Top management buy in
Org. Culture:				
Culture Orientation	Hierarchy	Hierarchy	Developmental	Development al
Structure	Functional	Weak Matrix	Strong Matrix	Projectized
Culture X agile	Departmental siloes; Conflict between corporate culture and highly collaborative, team oriented work processes	Departmental siloes; Insulated, dedicated team formed a separate subculture, which facilitated the methodology	Project work structure facilitated collaborative, team oriented methodology	Project work structure facilitated collaborative, team oriented methodology ; methodology training accelerated adoption
Development Scope	In-house	In-house	External Customer	External Customer
Industry	Healthcare	Healthcare	Software Development	Software Development
Agile Technique	Waterfall/Scrum Blend	Scrum	XP	Scrum

CONCLUSION

The goal of this study was to better understand the critical factors that affect the routinization of ASD methods in organizations. Particularly, we sought to investigate the acceptance-routinization gap in ASD assimilation through an exploratory case study to answer our research question: how do organizational factors (culture and structure) affect the routinization of ASD practices in organizations? Our purpose was to provide clarity

concerning the organizational factors that influence the routinization of ASD methods. We accomplished this by a careful empirical examination of four ASD assimilators, providing a rich contextual analysis of the complex issues involved in the assimilation process. As a result, this study makes several contributions to the existing literature on software development, OC, and IA.

First, our study adds to the literature on ASD assimilation by systematically examining the three assimilation gaps. As the literature calls for, this study provides a novel understanding of ASD assimilation beyond the acceptance stage. Intentionally, this study provides an in depth examination of the acceptance-routinization gap from real-life cases. In this study, perceived consequences of the continued use of ASD methods emerged as a critical factor in predicting the increase in the extent of adoption decision as shown in Figure 3.4. With regards to ASD assimilation, perceived consequences of future use yields helpful insight into the way stakeholders interpret ASD diffusion within a particular context. This link was found to be more closely tied to routinization decisions than project outcomes. This insight emerged as a result of investigating the acceptance-routinization gap in depth. Overall, these results fill knowledge gaps and provide insights to both academic and practitioner audiences.

*= emerging factor

Figure 3.4. Emergent Construct
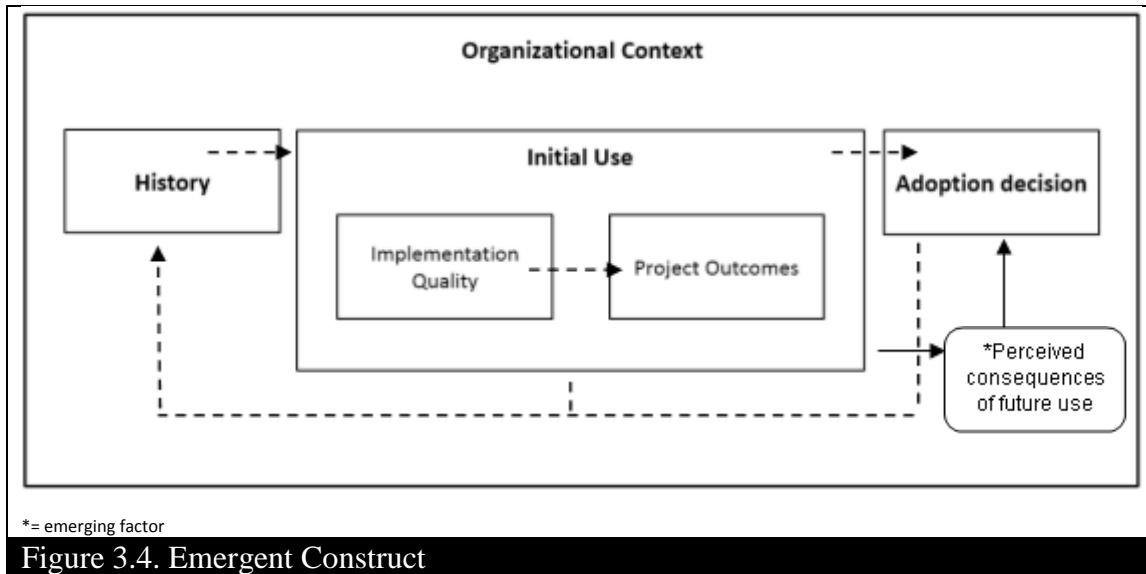
Second, our study adds to the literature on OC and ASD. Our results clarify the role of distinct OC types, hierarchical and developmental, as facilitators and inhibitors to ASD routinization intentions. Our empirical findings suggest that hierarchical cultures produce OSs that inhibit the long-term and widespread use of ASD methods. Using the CVM, we show how the values of a bottom-up method such as ASD methods, conflict with top-down decision making structures that are often present in hierarchical cultures. Nevertheless, organizations with hierarchical cultures can achieve success short-term success by creating a collaborative subculture by insulating the team from the presiding OC. However, in order to achieve long-term use of ASD, organizations with hierarchical cultures may be required to alter their existing OC and structure to facilitate ASD methods; an effort that may pose great difficulty. On the other hand, developmental cultures were found to facilitate ASD assimilation. As a result, ASD project success led to future use in developmental cultures, whereas hierarchical structure inhibited future use in hierarchical cultures; even after project success in hierarchical cultures.

Third, the findings of our study suggest that the relationship between OC and ASD assimilation is mutually reinforcing. As combined results suggests, an organizations willingness and ability to alter its OS to accommodate ASD methods may be impacted by both the perception of project success and consequences of continuance. This oscillating relationship may go through many iterations before ASD methods are finally routinized or rejected as a result of one or more of these factors. Thus, we propose an early model of ASD assimilation as shown in Figure 3.5.

Figure 3.5. Agile Assimilation Model
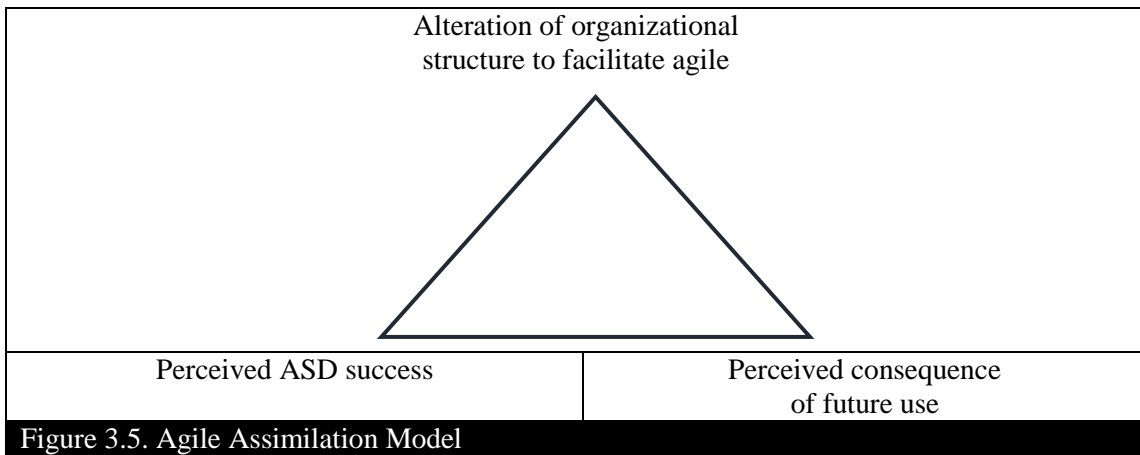
Practical Implications and Future Research

The results of this study yield significant implications for practitioners interested in the gaining the benefits from the normal use of ASD methods. In particular, by providing insights on ASD routinization, the results of this study build toward an understanding of ASD transformation, which may hold further insights on firm agility. In this study, we observed that the local application of ASD methods often brought local success. However, our results also suggest that the wider and deeper use of ASD principles throughout an organization as facilitated through a firm's OC, may lead to a higher degree of firm agility. Prior research also suggests that development cultures may lead to

higher degrees of firm agility than other culture types (Iivari and Iivari 2011). Both prior research and practitioner experiences have documented the benefits of ASD use, even within a company whose primarily focus is not software development.

Our findings suggest that firms that gain benefits from the local use of ASD methods that also want to implement ASD methods throughout, must alter their OC and corresponding OS to facilitate ASD processes. We call out future research to explore the role of culture and other enablers of ASD transformation. Understanding not only the benefits of ASD, but also how to transform an organizational from non-agile to agile would yield powerful implications for companies of all sizes.

Limitations

Our study, like all, has a few limitations. First, we studied four cases. Therefore, we call for future research to test the consistency and generalizability of the results found in this study. Second, since our goal was to study companies at different stages of ASD assimilation, some of our respondents shared most past experiences while others shared present experiences. In the case of Alpha, some of the company's developers had already left the company during this research. Third, our level of access varied depending on the company's willingness to discuss its ASD implementations, the availability of its stakeholders, and their current attitude towards the method. In cases where access was somewhat limited, we sought key informants to provide different perspectives on the company and its experiences with ASD methods. Fourth, three of our four cases describe ASD activities in co-located teams. Only Dynamic, included distributed ASD teams, in

which developers and quality assurance, project managers, and clients were in separate locations. Future research may investigate differences in the influence of organizational factors in ASD routinization between co-located and distributed ISD environments.

Despite these limitations, the researchers were able to fulfill the objectives of the study, answer the research question, and provide rich insights concerning ASD routinization. We hope that this work helps readers and researchers in understanding this phenomenon, and points the direction for future research.

REFERENCES

Abrahamsson, P., Conboy, K., and Wang, X. 2009. ""Lots done, more to do": the current state of agile systems development research," European Journal of Information Systems (18), pp 281-284.

Ambler, Scott. "Agile Adoption Survey Results." Survey. Accessed online. 11 June 2014.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., and Jeffries, R. 2001. "The agile manifesto".

Boehm, B., and Turner, R. 2005. "Management challenges to implementing agile processes in traditional development organizations," Software, IEEE (22:5), pp 30-39.

Cameron, K. S., and Quinn, R. E. 2011. "Diagnosing and changing organizational culture: Based on the competing values framework".

Chan, F. K., and Thong, J. Y. 2009. "Acceptance of agile methods: A critical review and conceptual framework," Decision Support Systems (46:4), pp 803-814.

Conboy, K. 2009. "Agility from first principles: reconstructing the concept of agility in information systems development," Information Systems Research (20:3), pp 329-354.

Cooper, R. B., and Zmud, R. W. 1990. "Information technology implementation research: a technological diffusion approach," Management science (36:2), pp 123-139.

Dubé, L. 1998. "Teams in packaged software development: The software corp. Experience," Information Technology & People (11:1), pp 36-61.

Dubé, L., and Robey, D. 1999. "Software stories: three cultural perspectives on the organizational practices of software development," Accounting, Management and Information Technologies (9:4), pp 223-259.

Fichman, R. G., and Kemerer, C. F. 1999. "The illusory diffusion of innovation: An examination of assimilation gaps," Information Systems Research (10:3), pp 255-275.

Hoffman, N., and Klepper, R. 2000. "Assimilating new technologies: The role of organizational culture," Information Systems Management (17:3).

Hofstede, G. 1980. Culture's consequences: International differences in work-related values. Sage Publications, Beverly Hills, CA, pp. 335-355.

Hong, W., Thong, J. Y., Chasalow, L. C., and Dhillon, G. 2011. "User acceptance of agile information systems: a model and empirical test," Journal of Management Information Systems (28:1), pp 235-272.

Iivari, J., and Huisman, M. 2007. "The relationship between organizational culture and the deployment of systems development methods," MIS Quarterly), pp 35-58.

Iivari, J., and Iivari, N. 2011. "The relationship between organizational culture and the deployment of agile methods," Information and Software Technology (53:5), pp 509-520.

Kitchell, S. 1995. "Corporate culture, environmental adaptation, and innovation adoption: a qualitative/quantitative approach," Journal of the Academy of Marketing Science (23:3), pp 195-205.

Kwon, T. H., and Zmud, R. W. 1987. "Unifying the fragmented models of information systems implementation," Critical issues in information systems research), pp 227-251.

Larson, E. W., and Gray, C. F. 2010. Project Management: The Managerial Process with MS Project. McGraw-Hill.

Leidner, D. E., and Kayworth, T. 2006. "Review: a review of culture in information systems research: toward a theory of information technology culture conflict," MIS quarterly (30:2), pp 357-399.

Mangalaraj, G., Mahapatra, R., and Nerur, S. 2009. "Acceptance of software process innovations–the case of extreme programming," European Journal of Information Systems (18:4), pp 344-354.

Meyer, A. D. and Goes, J. B. 1988. "Organizational assimilation of innovations: A multilevel contextual analysis," *Academy of management journal* (31:4), pp. 897-923.

Nerur, S., and Balijepally, V. 2007. "Theoretical reflections on agile development methods," Communications of the ACM (50:3), pp 79-83.

Nerur, S., Mahapatra, R., and Mangalaraj, G. 2005. "Challenges of migrating to agile methods," Communications of the ACM (48:5), pp 72-78.

Newman, M., and Robey, D. 1992. "A social process model of user-analyst relationships," MIS Quarterly, pp 249-266.

Overhage, S., and Schlauderer, S. Year. 2012 "Investigating the long-term acceptance of agile methods: An empirical study of developer perceptions in scrum projects," *45th Hawaii International Conference on System Science (hicss)*, IEEE2012, pp. 5452-5461.

Overeem, B. (2014, May 12). Lessons Learned as an Agile Coach. Retrieved from http://www.scrum.nl/prowareness/website/scrumblog.nsf/dx/lessons-learned-as-an-agile-coach.

Purvis, R. L., Sambamurthy, V., and Zmud, R. W. 2001. "The assimilation of knowledge platforms in organizations: An empirical investigation," *Organization science* (12:2), pp. 117-135.

Quinn, R. E. 1988. Beyond rational management: Mastering the paradoxes and competing demands of high performance, Jossey-Bass, San Francisco.

Quinn, R. E., and McGrath, M. R. 1985. "The transformation of organizational cultures: A competing values perspective," Organizational culture, pp 315-334.

Robinson, H., and Sharp, H. 2005. "Organisational culture and XP: three case studies," *Agile Conference, Proceedings*, pp.49, 58.

Rogers, E. M. 2003. *Diffusion of Innovations*, 5th ed., New York: Free Press.

Ruppel, C. P., and Harrington, S. J. 2001. "Sharing knowledge through intranets: A study of organizational culture and intranet implementation," Professional Communication, IEEE Transactions on (44:1), pp 37-52.

Russo, N., Shams, S., and Fitzgerald, G. 2013. "A Tale Of Two Agile Implementations: A Cross-Case Exploratory Analysis,").

Schein, E. H. 1985. "How culture forms, develops, and changes," Gaining control of the corporate culture), pp 17-43.

Senapathi, M., Drury, M., and Srinivasan, A. Year. "Agile Usage: Refining a Theoretical Model," PACIS2013, p. 43.

Senapathi, M., and Srinivasan, A. 2012. "Understanding post-adoptive agile usage: An exploratory cross-case analysis," Journal of Systems and Software (85:6), pp 1255-1268.

Senapathi, M., and Srinivasan, A. 2014. "An empirical investigation of the factors affecting agile usage," *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ACM2014, p. 10.

Strode, D. E., Huff, S. L., and Tretiakov, A. 2009. "The impact of organizational culture on agile method use," *42nd Hawaii International Conference on System Sciences*, IEEE2009, pp. 1-9.

Tolfo, C., and Wazlawick, R. S. 2008. "The influence of organizational culture on the adoption of extreme programming," Journal of Systems and Software (81:11), pp 1955-1967.

Version One. "8th Annual State of Agile Survey." Survey. Accessed online. 15 June 2014.

Vidgen, R., and Wang, X. 2009. "Coevolving systems and the organization of agile software development," Information Systems Research (20:3), pp 355-376.

Vijayasarathy, L., and Turk, D. 2012. "Drivers of agile software development use: Dialectic interplay between benefits and hindrances," Information and Software Technology (54:2), pp 137-148.

Wang, X., Conboy, K., and Pikkarainen, M. 2012. "Assimilation of agile practices in use," Information Systems Journal (22:6), pp 435-455.

Weigel, F., Hazen, B., Cegielski, C., and Hall, D. 2014. "Diffusion of Innovations and the Theory of Planned Behavior in Information Systems Research: A Metaanalysis," *Communications of the Association for Information Systems* (34:1), p. 31.

Yin, R. K. 2008. *Case study research: Design and methods*.

CHAPTER 4: THE IMPACT OF ORGANIZATIONAL CULTURE AND STRUCTURE ON THE ROUTINIZATION OF AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

INTRODUCTION

The *agile software development* (ASD) practice can be defined as a software development team's ability to respond to changing user requirements through a process of continual readiness (Conboy 2009; Lee and Xia 2010). ASD adoption rates both within large organizations and across the software development (SD) industry have seen significant growth since the signing of the *Manifesto for Agile Software Development* in 2001 (Beck et al. 2001; Murphy et al. 2013). For instance, one survey reports that companies that plan to implement ASD for future development projects increased from 59 percent in 2011 to 83 percent in 2012 (Version One 2013). Ironically, some organizations are electing to discontinue their use of ASD methods shortly after experiencing the benefits of ASD use (Denning 2012a, 2012b; Sutherland 2014; Version One 2015). Thus, the assumption that successful ASD use will lead to further ASD use has not held. This begs the question: why are some organizations not using ASD methods routinely after achieving ASD project success?

The extant literature underscores the reasons why ASD methods are initially adopted (i.e. comparative advantage, etc.) (Rogers 2003; Tessem 2003; Bahli and Zeid 2005; Senapathi and Srinivasan 2012), however, there is a dearth of research concerning the routinization of ASD in organizations despite calls for a greater understanding of ASD

adoption beyond the initial stages (Abrahamsson et al. 2009). The literature lacks an understanding of the factors that separate those adopters that choose to use ASD methods on a routine basis from those adopters that forego their use of ASD after experiencing the promised benefits of ASD methods (e.g. faster time to market, higher customer satisfaction, etc.). Therefore, we draw upon diffusion of innovation theory (DOI) to understand the gap between the perceived success of an initial ASD use and ASD routinization. Such insight is critical as more organizations pour resources into making ASD their principal method of software development. As a theoretical lens, diffusion of innovation theory (DOI) examines the extent that ASD represents an innovation in terms of a fundamental shift in the way that software development has typically been done. In line with prior research, we use DOI to provide some insight into the progression in which an innovation proliferates throughout an organization.

Prior research suggests that organizational factors play a pivotal role in the diffusion of an innovation within an organization (Chan and Thong 2009). In particular, agile's emphasis on social interactions and continual readiness towards changing requirements (Beck et al. 2001; Conboy 2009) has been shown to induce significant changes to more hierarchical organizational structures (Nerur 2005). Additionally, the inability to change an organization's culture has been identified as a key impediment to ASD assimilation (Ambler 2014; Version One 2014). Therefore, we employ organizational culture theory to explore the relationship between organizational factors of culture and structure, and their influence on the ASD routinization process.

In summary, this study seeks to contribute to the current body of ASD research by providing an understanding of gap between initial ASD success (acceptance) and ASD routinization. This gap motivated the following research question: *how do organizational factors of culture and structure affect the routinization of ASD practices in organizations?* Given the gaps in the literature, our study intends to evolve a theoretical model that explains how organizational factors such as organizational culture and organizational structure influence the ASD routinization process, given the success of initial ASD use. Our purpose is to provide clarity concerning the organizational factors that influence the routinization of ASD methods. To accomplish this, we conceptualize a parsimonious model that represents relationships among related constructs, with rich theoretical explanations from the extant literature.

The remainder of the article proceeds as follows. First, the theoretical foundations section provides an overview of past research and relevant literature on the theories used to answer our research question. Second, the hypothesis section will layout the theoretical relationship represented in our model. Third, the discussion section will elaborate on the research strategy and the implications of our study on future research.

LITERATURE REVIEW

In this section, we give an overview of the research on innovation theory and organizational culture theory, which were used as the basis for this work.

Innovation Assimilation

The extant literature extends diffusion of innovation theory (DOI) (Rogers 1962) from individual to organizational-level adoption (Rogers 2003) to study the diffusion of complex innovations in organizations (Kwon and Zmud 1987; Meyer and Goes 1988; Cooper and Zmud 1990; Gallivan 2001). We refer to the research on organizational-level assimilation of innovations as *assimilation theory* (Wang et al. 2012). The research model developed by Kwon and Zmud (1987) and further refined by Cooper and Zmud (1990) defines *six-stages of innovation assimilation* that describe technology implementations in organizations. *Innovation assimilation (IA)* is defined as the extent to which the use of an innovation diffuses across "the organizational projects or work processes and becomes routinized in the activities of those projects and processes" (Purvis, Sambamurthy, and Zmud 2001). Table 4.1 presents the six-stages of the IA model, defining each stage of the process and delineating the adoptive use stages from the post-adoptive use stages. Each stage describes a differing level at which an innovation diffuses an adopting unit (Overhage and Schlauderer 2012). Like Roger's (1962) original DOI model, Kwon and Zmud's (1987) six-stage IA model has been adapted to study not only technology innovations, but also ideas and practices. ASD methods can be considered an SD innovation (Chan and Thong 2009) because they represent a significant departure from previous methods for building software, as determined by the collective judgements of experts in the field (Meyers and Goes 1988; Beck et al. 2001). Moreover, the implications of studying ASD as an innovation and not just as a method potentially yield insights as to the broader impacts of ASD implementations across organizations (Wang et al. 2012; Senapathi and Srinivasan 2012).

Table 4.1. Definitions of the Assimilation Stages (Cooper and Zmud; Senapathi and Srinivasan 2012)
Innovation Assimilation Stages

Adoptive Stages
1. Initiation: need for change is recognized, a match is identified between an innovation and its application in the organization
2. Adoption: a decision is made to adopt an innovation
3. Adaptation: an adaptation to suit the contextual needs

Post-adoptive Stages
4. Acceptance: use of the innovation
5. Routinization: an increase in the extent and intensity of use (i.e. usage of the innovation is encouraged as a normal activity)
6. Infusion: increased usage in a more comprehensive and integrated manner results in increased effectiveness of systems development (i.e. the innovation penetrating deeply into an organization).

ASD assimilation

Because our focus is on the organization-level, we draw upon IA research to study the diffusion of ASD methods in organizations (Kwon and Zmud 1987). We define *ASD assimilation* as the extent to which the use of ASD methods diffuse across the organizational projects or work processes and becomes routinized in the activities of those projects and processes (Cooper and Zmud 1990; Wang et al. 2012). With respect to Kwon and Zmud's (1987) six-stage IA model, the early ASD literature focuses primarily on understanding the adoptive use stages (initiation, adoption, and adaption) of ASD assimilation (Tessem 2003; Bahli and Zeid 2005; Svensson and Höst 2005; Nerur et al. 2005), while later studies provide insights at the post-adoptive use stages (acceptance, routinization, infusion) (Chan and Thong 2009; Mangalaraj et al. 2009; Vijayasarathy and Turk 2012; Overhage and Schlauderer 2012) (see Table 4.1). The studies that apply innovation theories at the post-adoptive agile use stages largely aim to understand the antecedents that affect agile use (acceptance). These antecedents differ depending on the

theoretical perspective and level of analysis employed by the researchers. Despite their differences, these studies broadly agree that: 1) the antecedents that drive an innovation across the adoptive stages differ from those that impact the post-adoptive stages, and 2) further research of the downstream phases of ASD assimilation is needed to understand how to sustain agile use long-term.

Additionally, past research contains models that explain the impact of deep ASD usage on outcomes (Chan and Thong 2009; Mangalaraj et al. 2009; Senapathi and Srinivasan 2012; Senapathi and Srinivasan 2014), but these models do not fully explain how the transition is made from one post-adoptive use phase to another at the organizational-level. These models do not specifically draw out contextual factors that impact ASD assimilation such as hierarchy and bureaucracy (Russo, Shams, and Fitzgerald 2013). With respect to Kwon and Zmud's (1987) six-stage IA model, there is a need for examination of the differences between those organizations that elect to routinize ASD methods from those that elect not to routinize after experiencing success at the acceptance stage. The extant literature is missing an understanding of *how* the transition from the acceptance to routinization stages occurs and *why* some organizations elect to adopt agile methodologies, not only initially, but also on a continual basis. As a result, there is a need for understanding the factors that affect the routinization decisions after initial use.

Using the research on assimilation gaps, we can complement the IA model by studying the forces that influence the diffusion between assimilation stages. *Assimilation gaps*

refer the difference between the patterns of cumulative assimilation events of an innovation across a population of adopters (Fichman and Kemerer 1999). Indeed, one of the limitations of the stage models upon which the innovation theories are built is the neglect of details between each stage (Sabherwal and Robey 1995). As a result, "the ASD assimilation process illustrates the path of assimilation, but it cannot answer, at least by itself, questions such as 'how and why the assimilation of practice progress from one stage to another?'" (Wang et al. 2012). According to the research on assimilation gaps, substantial gaps can present a misleading image of the diffusion process, which leads to inaccurate conclusions about the strength of the diffusion process that is being observed. Consequently, erroneous theoretical and practical inferences may be drawn based on false assumptions (Fichman and Kemerer 1999). For instance, using the IA model by itself, one might assume that an innovation that successfully diffuses through one assimilation stage will automatically advance to the next. However, regarding the transition from acceptance to routinization, practitioner literature suggests that some organizations elect not to routinize ASD methods after successfully completing the acceptance stage (Denning 2012a, 2012b; Sutherland 2014). We define *ASD acceptance* as the introductory adoption and employment of ASD methods as a process innovation for one or more software development projects (Vijayasarathy and Turk 2012; Wang et al. 2012). Similarly, we define *ASD routinization* as the usage of ASD methods as a normal activity in the organization, where the innovation is no longer considered out of the ordinary (Wang et al. 2012).

Thus, this study combines the innovation theories and the research on assimilation gaps to understand the process and the factors that impact the ASD assimilation process in organizations. The objective of this study is to provide an understanding of the organizational factors that influence ASD routinization after initial ASD use. To accomplish this, we leverage the strengths of the IA research –- to examine the concept of *use*, not in a binary perspective (i.e. use vs. non-use), but rather the extent to which an innovation is used and how its use influences an organization's practices, structures, and organizational culture (Gallivan 2001) --- to study the transition from successful ASD acceptance to ASD routinization.

Organizational culture theory

To investigate the relationship between OC and post-adoptive ASD use, we draw on the organizational culture (OC) literature to highlight the composition of OC, the different OC orientations, and the effect of OC on the use of ASD methods. Although organization culture's broad conceptualization makes it arguably connected to every organizational process, OC's complex, interrelated, and somewhat ambiguous set of factors make it impossible to create a comprehensive framework (Cameron and Quinn 2011). Consequently, researchers have had marginal success integrating and organizing elements of OC into widely used frameworks. One exception is Schein's (1985) three-layer model of basic assumptions, espoused values, and artifacts, which conceptualizes observable and unobservable layers of OC. Using Schein's (1985) definition of group culture, we define OC as "a pattern of shared basic assumptions learned by an [organization] as it solves its problems of external adaptation and internal integration, which has worked well

enough to be considered valid and, therefore, to new members as the correct way to perceive, think, and feel in relation to those problems". His three-layer model suggests that aspects of OC can be tapped by focusing on manifesting elements that are theoretically and practically tied to the unobservable layers. Thus, the vast majority of OC studies focus on values as constituents of OC (Hofstede 1980; Quinn and McGrath 1985).

Following prior SD research (Iivari and Huisman 2007), we draw upon the Competing Values Model (CVM) to conceptualize OC as the manifestation of competing value systems. These culture types form from four quadrants, differentiated by the continuums of two dimensions, internal to external focus and change to stability as shown in Figure 4.1 (Iivari and Huisman 2007). Each of the four quadrants in the CVM represents a distinct OC, which we will draw upon to differentiate opposing OCs according to their core of values. We use Iivari and Huisman's (2007) categorizations of OC as shown below.

- *Group culture* (change and internal focus) is primarily concerned with human relations and flexibility. Belonging, trust, and participation are its core values. Effectiveness criteria include the development of human potential and member commitment.

- *Developmental culture* (change and external focus) is future-oriented, considering what might be. The effectiveness criteria emphasize growth, resource acquisition, creativity and adaptation to the external environment.

- *Hierarchical culture* (stability and internal focus) is oriented toward security, order, and routinization. It emphasizes control, stability and efficiency through the following of regulations.

- *Rational culture* (stability and external focus) is achievement-oriented, focusing on productivity, efficiency, and goal achievement.

Figure 4.1. Competing Values (Iivari and Huisman 2007)

In our research, we focus specifically on the interaction of ASD methods and hierarchical and developmental OCs for the following reasons: First, our primary proposition is that ASD values compete and conflict with those of a hierarchical OC. Thus, a hierarchical OC should have a negative influence on ASD routinization. Given that the hierarchical and developmental OCs are polar opposites in the CVM framework, these OC types pose competing and conflicting demands on organizations (Iivari and Huisman 2007). Second, from a competing values standpoint, the emphasis of the developmental OC closely aligns with the tenets of the ASD philosophy as espoused in the Agile Manifesto. Thus, we chose a more parsimonious approach to test our hypothesis by focusing on the hierarchical and development cultures to discriminate between OCs based on their most salient differences. These differences manifest themselves in an organization's structure, which we will discuss in the next section.

Organizational Structure

Organizational structure (OS) can be defined as "the logically consistent clustering of an organization's elements as it searches for harmony in its internal processes and consonance with its environment" (Mintzberg 1980). According the project management body of knowledge (Rose 2013), organizational structures range from functional to projectized, and can affect the availability of resources and influence how projects are conducted (Rose 2013). With respect to prior research, we postulate that an organization will produce an organizational structure that is reflective of its OC orientation (Martin 1992; Sackmann 1992). Second, we postulate that the relationships between OS and ASD routinization will parallel the relationship between OC and ASD, as OS is a manifestation of OC. Thus, we match the OC orientations with their corresponding OS as follows: hierarchical OC to functional OS and developmental OC to projectized OS. The Rose (2013) defines these structures as shown below:

- *Functional OS* is a hierarchical organization where each employee has one clear superior, and staff are grouped by areas of specialization and managed by a person with expertise in that area.
- *Projectized OS* is any organizational structure in which the project manager has full authority to assign priorities, apply resources, and direct the work of persons assigned to the project.

Table 4.2 summarizes that difference between the hierarchical and developmental OCs, and their resulting organizational structures base   d on the prior research of Russo, Shams, and Fitzgerald (2013), Iivari and Huisman (2007), and Gallivan (2001).

Table 4.2. A Comparison of Hierarchical and Developmental organizational cultures		
Organizational Culture	Hierarchy	Developmental
Organizational Strategy	Specified and predictable	Flexibility and adaptive
Enterprise Governance Structure	Bureaucratic	Adhocracy
Team Management Structure	Functional	Projectized
Decision Making Structure	Top-down	Bottom-up
Team Work Structure	Siloed	Collaborative
Control Mode	Formal	Informal
Organizational Focus	Internal	External
Stability vs. Change	Stability	Change

The Impact of Organizational Culture on ASD

Prior research has shown interest in the relationship between cultural values and SD process improvement. For example, Dubé and colleagues studied the compatibility between organizational values and values subgroups (Dubé 1998, Dubé and Robey 1999) with results suggesting that high compatibility between the organizational values and a given process innovation, the more successful the implementation is likely to be. More recently, the SD researchers have studied the relationship between OC and the deployment of SD methods (Iivari and Huisman 2007). Regarding ASD methods, most researchers have focused on the relationship between OC and ASD use (Robinson and Sharp 2005, Tolfo and Wazlawick 2008, and Strode et al. 2009), which is usually studied in the early adoptive stages or simply use versus non-use, with few exceptions (Iivari and Iivari 2011). However, despite calls for future empirical investigation on the influence of OC on the later stages of ASD assimilation, (Vijayasarathy and Turk 2012; Wang et al. 2012) there remains only a dearth of examination on this relationship. Additionally, IA research suggests that OC plays a key role in the continual use of an innovation, however, no studies were found that explore the relationship between OC and ASD routinization

specifically. Therefore, we study the influence of organizational factors (culture and structure) on the routinization of ASD methods with the intention of theory building. Our model, as depicted in Figure 4.2, describes a major gap between the perceived success of an initial ASD use and ASD routinization, that being the role of organizational factors, culture and structure. Although we recognize that the relationship between OC and OS on ASD use is not one way, the model is useful in highlighting the way in which organizational factors play a pivotal role in the ASD routinization process.
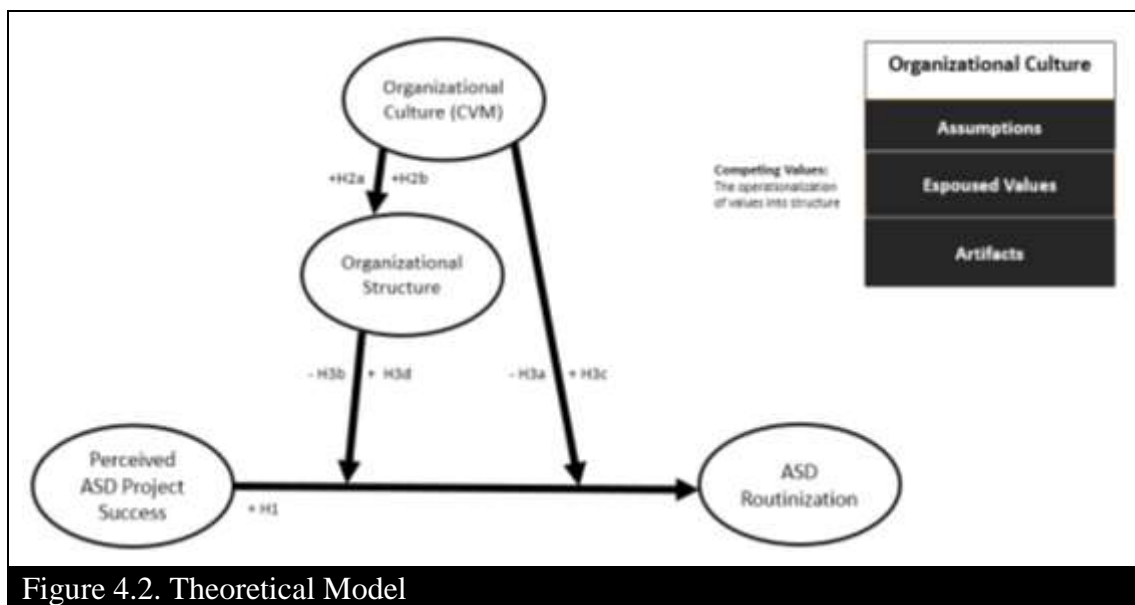
Figure 4.2. Theoretical Model

In the next section, we describe the constructs in our model --- organizational culture, organizational structure, initial agile usage, perceptions of project success, and ASD routinization, and then test our hypotheses as shown in Table 4.3 concerning the relationships therein.

RESEARCH MODEL AND PROPOSITIONS

Proposition 1: A firm's perceived ASD success will influence ASD routinization.	H1: Perceived ASD project success will positively influence ASD routinization.
Proposition 2: A firm's organizational culture, represented by its core values, will influence its organization structure (form).	H2a: A hierarchical organizational culture will give rise to a functional organizational structure H2b: A developmental organizational culture will give rise to a projectized organizational structure.
Proposition 3: Organizational factors, culture and structure, will influence ASD routinization beyond the perceived ASD project success.	Negative Influence H3a: A hierarchical organization culture will negatively moderate the influence of perceived ASD project success on ASD routinization. H3b: A functional organization structure will negatively moderate the influence of perceived ASD project success on ASD routinization. Positive Influence H3c: A developmental organization culture will positively moderate the influence of perceived ASD project success on ASD routinization. H3d: A projectized organization structure will positively moderate the influence of perceived ASD project success on ASD routinization.

Table 4.3. Propositions and Hypothesis

Perceived ASD Project Success and ASD Routinization

Perceived ASD project success is the extent to which an ASD project meets technical goals, remains within the budget, is delivered in time, and is accepted by the end user (Jiang, Klein, and Pick 2003; Procaccino and Verner 2006). Both the increase in ASD adoption rates and the findings of empirical research suggest that the use of ASD methods yields a number of comparative advantages over traditional methods such as higher team productivity (Layman et al. 2004), higher customer satisfaction (Ceschi et al. 2005), more efficient requirements gathering (Hansen and Lyytinen 2014), and a greater ability to handle changing user requirements (Vigden and Wang 2009), with few studies

reporting contrary results (Dalcher et al. 2005). Furthermore, the use of ASD methods has been associated with a higher rate of project success and in turn, a lower rate of project failure (Chaos 2010). From an organizational perspective, the success of ASD projects has been determined by not only traditional factors of time, scope, and budget (Keider 1974; Boehm, 1981; Pinto and Slevin 1988), but also factors such as meeting user requirements (Procaccino and Verner 2006).

ASD routinization refers to the usage of ASD methods as a normal activity in the organization. During this stage, ASD methods are no longer considered out of the ordinary (Wang et al. 2012). Prior research suggests that the impacts of using a particular software development methodology will influence developers' attitudes towards the future use of the methodology (Khalifa and Verner 2000). Prior research also suggests that developers may be more apt to use a methodology if he or she perceives that they are more effective when using the methodology (Green and Hevner 1999). Thus, it follows that a firm's perceived ASD success during ASD acceptance will positively influence its attitudes toward ASD routinization.

Proposition 1: A firm's perceived ASD success will influence ASD routinization

On the other hand, both research and practice challenge this assumption by showing those organizations that do not routinize ASD methods even after experiencing benefits of agile use. In many cases, the decision concerning whether an organization, department, or team decides to continue using ASD methods is determined by compatibility of the methodology with the organizational culture and social norms (Dubé and Robey 1999;

Sutherland 2014). Thus, we investigate the influence of organizational factors of culture and structure on the relationship between perceived ASD success and ASD routinization.

Organizational Culture, Organizational Structure, and ASD Routinization

Prior research has emphasized the importance of cultural compatibility in relationship to a given IS effort (Dubé 1998). That withstanding, the compatibility and relative fit of a top-down, hierarchical organizational culture to a bottom-up methodology like agile, has been scrutinized (Boehm and Turner 2005; Nerur and Balijepally 2007). Using the CVM, we propose that a hierarchical organizational culture (Quinn 1988) predicated on bureaucracy, should produce organizational structures to support the culture (H2a). On the other hand, we propose that developmental organizational cultures (Quinn 1988) are predicated on that these bottom-up cultures should produce organizational structures to support the culture.

Proposition 2: A firm's organizational culture, represented by its core values, will influence its organization structure (form).

Prior research points out that change to both the organization and the innovation may occur in order to exploit the innovation (Rogers 1983; Gallivan 2001). During the implementation phase, an innovation may conflict with sources of structure and social norms within an organization, which may lead to the rejection of a particular innovation. On the other hand, when ASD acceptance leads to perceive ASD success, this relationship may be assumed to lead to the routinization of ASD methods in all cases. However, we propose that organizational factors, particularly culture and structure,

significantly influence the relationship between perceived ASD success and ASD routinization. (P3). Thus,

> Proposition 3: Organizational factors, culture and structure, will influence ASD routinization beyond the perceived ASD project success.

Furthermore, the process of adapting an innovation includes redefining and restructuring organizational processes, so that members understand the innovation and its role in meeting organizational goals (Gallivan 2001). Therefore, we hypothesize that when organizational values are in competition with agile values, organizational factors of culture and structure may inhibitor the assimilation of ASD methods (H3a, H3b). On the other hand, when organizational values align with ASD values, organizational factors, culture and structure, may facilitate the assimilation of ASD methods (H3c, H3d). In this study, we posit that developmental OCs are more compatible with ASD methods than hierarchical OCs, which negatively influence ASD routinization (Iivari and Iivari 2011).

CONCLUSION

The goal of this study was to better understand the critical factors that affect the routinization of ASD methods in organizations. Particularly, we sought to understand the gap between the perceived success of an initial ASD use and ASD routinization to answer our research question: *how do organizational factors (culture and structure) affect the routinization of ASD practices in organizations?* Our purpose was to provide clarity concerning the organizational factors that influence the routinization of ASD methods.

We accomplished this by evolving a theoretical model from the extant literature. Our study adds to the literature on ASD assimilation by providing insights concerning a key ASD assimilation gap. As the literature calls for, this study provides a novel understanding of ASD assimilation beyond the acceptance stage.

Limitations and Future Research

A clear limitation of the study is that the model has not been empirically validated. Future research is needed to test the relationships proposed in the model.

As set forth in prior research, defining culture poses many challenges to researchers. Although culture has commonly been conceptualized as consisting of multiple levels, organizational culture is still difficult to define, conceptualize, and operationalize with certainty. Our work follows prior research, which draws upon the CVM model while accepting that there are other ways to study culture.

In our study, we aim to develop a parsimonious model to explicate the impact of key organizational factors on ASD use. As a consequence, our model does not elaborate on the recursive relationships that may exists between constructs. We conceptualize factors that influence ASD use, but we do not have room to discuss how ASD use impacts these factors in reciprocally. Future research is needed to further investigate these mutual impacts.

The practical contributions of this research are listed as follows. First, our model sets forth a set of propositions that can aid organizations in understanding ASD routinization issues. Particularly, the impact of an organization's culture on specific factors can be gained from this research. Whereas past research and practitioner surveys have proposed the role of organizational culture in the ASD process, there is little research that has sought to explain how organizational culture might affect the routinization process. Once operationalized, our model may break new ground in the area of ASD assimilation research. Furthermore, understanding ASD routinization holds significant impact for understanding agile transformation, or the organizational change from less agile to more agile, as the ASD assimilation process is key in an organization's pursuit of firm agility.

Despite these limitations, the researchers were able to fulfill the objectives of the study, answer the research question, and provide rich insights concerning ASD routinization. We hope that this work helps readers and researchers in understanding this phenomenon, and points the direction for future research.

REFERENCES

Abrahamsson, P., Conboy, K., and Wang, X. 2009. ""Lots done, more to do": the current state of agile systems development research," *European Journal of Information Systems* (18), pp 281-284.

Ambler, Scott. "Agile Adoption Survey Results." Survey. Accessed online. 11 June 2014.

Bahli, B. and Zeid, E.A. 2005. "The role of knowledge creation in adopting extreme programming model: an empirical study," In *Information and Communications Technology. Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference on*, pp. 75-87.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., and Jeffries, R. 2001. "The agile manifesto".

Boehm, B., and Turner, R. 2005. "Management challenges to implementing agile processes in traditional development organizations," *Software, IEEE* (22:5), pp 30-39.

Cameron, K. S., and Quinn, R. E. 2011. "Diagnosing and changing organizational culture: Based on the competing values framework".

Chan, F. K., and Thong, J. Y. 2009. "Acceptance of agile methods: A critical review and conceptual framework," *Decision Support Systems* (46:4), pp 803-814.

Chaos, tech. report 2010. Standish Group Int'l.

Conboy, K. 2009. "Agility from first principles: reconstructing the concept of agility in information systems development," *Information Systems Research* (20:3), pp 329-354.

Cooper, R. B., and Zmud, R. W. 1990. "Information technology implementation research: a technological diffusion approach," *Management Science* (36:2), pp 123-139.

Dalcher, D., Benediktsson, O., and Thorbergsson, H. 2005. "Development life cycle management: a multiproject experiment," in *Engineering of Computer-Based Systems, ECBS'05. 12th IEEE International Conference and Workshops on the, IEEE,* pp. 289-296.

Denning 2012a. "The Best-Kept Management Secret On The Planet: Agile," *Forbes*. Accessed online. 1 September 2015.

Denning 2012b. "The Case Against Agile: Ten Perennial Management Objections," *Forbes*. Accessed online. 1 September 2015.

Dubé, L. 1998. "Teams in packaged software development: The software corp. Experience," *Information Technology & People* (11:1), pp 36-61.

Dubé, L., and Robey, D. 1999. "Software stories: three cultural perspectives on the organizational practices of software development," *Accounting, Management and Information Technologies* (9:4), pp. 223-259.

Fichman, R. G., and Kemerer, C. F. 1999. "The illusory diffusion of innovation: An examination of assimilation gaps," *Information Systems Research* (10:3), pp 255-275.

Gallivan, M. J. 2001. "Organizational adoption and assimilation of complex technological innovations: development and application of a new framework," *ACM Sigmis Database* (32:3), pp. 51-85.

Green, G. C. and Hevner, A. R. 1999. "Perceived Control of Software Developers and its Impact on the Successful Diffusion of Information Technology," *Carnegie Mellon, Software Engineering Institute*, Pittsburg, PA.

Hansen, S. and Lyytinen, K. 2014. "Requirements Computation: Analyzing Requirements Cognition in Multiple Development Paradigms," in *AMCIS*.

Hofstede, G. 1980. *Culture's consequences: International differences in work-related values.* Sage Publications, Beverly Hills, CA, pp. 335-355.

Iivari, J., and Huisman, M. 2007. "The relationship between organizational culture and the deployment of systems development methods," *MIS Quarterly*, pp. 35-58.

Iivari, J., and Iivari, N. 2011. "The relationship between organizational culture and the deployment of agile methods," *Information and Software Technology* (53:5), pp. 509-520.

Jiang, J. J., Klein, G., and Pick, R. A. 2003. "The impact of IS department organizational environments upon project team performances," *Information and Management* (40:3), pp. 213-220.

Keider, S. P. 1974. "Why projects fail," *Datamation* (20:12), pp. 53-55.

Khalifa, M. and Verner, J. 2000. "Drivers for Software Development Usage," *IEEE Transactions on Engineering Management* (47:3), pp. 360-369.

Kwon, T. H., and Zmud, R. W. 1987. "Unifying the fragmented models of information systems implementation," *Critical issues in information systems research*, pp. 227-251.

Layman, L., Williams, L., and Cunningham, L. 2004. "Exploring extreme programming in context: an industrial case study," *Agile Development Conference, 2004, IEEE*, pp. 32-41.

Lee, G., and Xia, W. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data," *MIS Quarterly* (34:1), pp. 87-114.

Mangalaraj, G., Mahapatra, R., and Nerur, S. 2009. "Acceptance of software process innovations–the case of extreme programming," *European Journal of Information Systems* (18:4), pp. 344-354.

Martin, J. 1992. *Cultures in Organizations: Three Perspectives*, Oxford, New York.

Meyer, A. D. and Goes, J. B. 1988. "Organizational assimilation of innovations: A multilevel contextual analysis," *Academy of management journal* (31:4), pp. 897-923.

Mintzberg, H. 1980. "Structure in 5's: A Synthesis of the Research on Organization Design," *Management science* (26:3), pp. 322-341.

Murphy, B., Bird, C., Nagappan, N., Zimmermann, T., Williams, L., and Begel, A. 2013. "Have Agile techniques been the silver bullet for software development at Microsoft?", in: *2013 ACM/IEEE international symposium on empirical software engineering and measurement*, p. 75–84. *http://ieeexplore.ieee.org/lpdocs/epic03/ wrapper.htm?arnumber=6681340*. Accessed 8 June 2014

Nerur, S., Mahapatra, R., and Mangalaraj, G. 2005. "Challenges of migrating to agile methods," *Communications of the ACM* (48:5), pp. 72-78.

Overhage, S., and Schlauderer, S. Year. 2012. "Investigating the long-term acceptance of agile methods: An empirical study of developer perceptions in scrum projects," *45th Hawaii International Conference on System Science (hicss)*, IEEE 2012, pp. 5452-5461.

Pinto, J. K., and Slevin, D. P. 1988. "Project success: definitions and measurement techniques," *Project Management Institute*.

Procaccino, J. D., and Verner, J. M. 2006. "Software project managers and project success: An exploratory study," *Journal of Systems and Software* (79:11), pp. 1541-1551.

Purvis, R. L., Sambamurthy, V., and Zmud, R. W. 2001. "The assimilation of knowledge platforms in organizations: An empirical investigation," *Organization science* (12:2), pp. 117-135.

Quinn, R. E. 1988. *Beyond rational management: Mastering the paradoxes and competing demands of high performance*, Jossey-Bass, San Francisco.

Quinn, R. E., and McGrath, M. R. 1985. "The transformation of organizational cultures: A competing values perspective*," Organizational culture*, pp 315-334.

Robinson, H., and Sharp, H. 2005. "Organisational culture and XP: three case studies," *Agile Conference, Proceedings*, pp. 49-58.

Rogers, E. M. 1962. *Diffusion of innovations*. Simon and Schuster.

Rogers, E. M. 2003. *Diffusion of Innovations*, 5th ed., New York: Free Press.

Rose, K. H., 2013. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition. *Project Management Journal*, (*44*:3), pp.e1-e1.

Russo, N., Shams, S., and Fitzgerald, G. 2013. "A Tale Of Two Agile Implementations: A Cross-Case Exploratory Analysis," in the *UK Academy for Information Systems Conference*.

Sabherwal, R. and Robey, D., 1995. "Reconciling variance and process strategies for studying information system development," *Information Systems Research*, (6:4), pp. 303-327.

Sackmann, S. A. 1992. "Culture and subcultures: An analysis of organizational knowledge," *Administrative science quarterly*, pp. 140-161.

Schein, E. H. 1985. "How culture forms, develops, and changes," *Gaining Control of the Corporate Culture*, R. H. Kilmann, M. J. Saxton, R. Serpa, and Associates (eds.), Jossey-Bass, San Francisco, pp. 17-43.

Senapathi, M., and Srinivasan, A. 2012. "Understanding post-adoptive agile usage: An exploratory cross-case analysis," *Journal of Systems and Software* (85:6), pp. 1255-1268.

Senapathi, M., and Srinivasan, A. 2014. "An empirical investigation of the factors affecting agile usage," *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ACM 2014, p. 10.

Strode, D. E., Huff, S. L., and Tretiakov, A. 2009. "The impact of organizational culture on agile method use," *42nd Hawaii International Conference on System Sciences*, IEEE2009, pp. 1-9.

Sutherland, J. 2014. *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Business.

Svensson, H. and Höst, M. 2005. "Introducing an agile process in a software maintenance and evolution organization," in *Software Maintenance and Reengineering, 2005. CSMR 2005. Ninth European Conference on*, pp. 256-264.

Tessem, B. 2003. "Experiences in learning XP practices: A qualitative study," in *Extreme Programming and Agile Processes in Software Engineering,* Springer Berlin Heidelberg, pp. 131-137.

Tolfo, C., and Wazlawick, R. S. 2008. "The influence of organizational culture on the adoption of extreme programming," Journal of Systems and Software (81:11), pp 1955-1967.

Version One. "7th Annual State of Agile Survey." Survey. Accessed online. 15 June 2013.

Version One. "8th Annual State of Agile Survey." Survey. Accessed online. 15 June 2014.

Version One. "9th Annual State of Agile Survey." Survey. Accessed online. 15 June 2015.

Vidgen, R., and Wang, X. 2009. "Coevolving systems and the organization of agile software development," *Information Systems Research* (20:3), pp 355-376.

Vijayasarathy, L., and Turk, D. 2012. "Drivers of agile software development use: Dialectic interplay between benefits and hindrances," *Information and Software Technology* (54:2), pp 137-148.

Wang, X., Conboy, K., and Pikkarainen, M. 2012. "Assimilation of agile practices in use," *Information Systems Journal* (22:6), pp 435-455.

CHAPTER 5: CONCLUSION

This research developed and tested a new model on the routinization of agile software methods in organizations using a three-essay format. Each essay has been developed for submission to high quality IS journals. Additionally, future work has been outlined as follows. Extensions to the first essay include expanding the thematic analysis to elaborate further about what is known versus what is not known. Secondly, I plan to refine the model to provide further explanations about how organizations factors specifically affect the unique elements of ASD methods. Extensions to the second essay include examining the mutually reinforcing relationship between OC and ASD methods. Secondly, I plan to examine the concept of agile transformation in light of the findings discussion in second essay. Extensions to the third essay include conducting a quantitative analysis on agile routinization using structural equation modeling, which builds on the first two essays.

APPENDIX A

APPENDIX A1

The Concept of Software Development Agility

Extant research indicates that the concept of agility first appeared in the mainstream business literature in the early 1990s (Goldman et al. 1991). Prior studies explored the concept of agility concerning manufacturing, management, product development, and other business research development (Takeuchi and Nonaka 1986; Sugimori 1977). Despite the contributions from these fields, the term "agile" became widely popular after the advent of the Agile Manifesto (see Table A1; Beck et al. 2001) in 2001, a document developed by group of software development practitioners that marked a new approach to building software.

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Figure A1. Agile Manifesto

Agile software development methodologies represent a departure from the traditional, waterfall approaches of building software according to iterative, incremental delivery

approaches. Agile is intended to be rapid as well as adaptive, highly collaborative, and team-oriented (Melão and Pidd 2000; McHugh 2012). Although the agile philosophy represents lean, lightweight development methods, it does not completely abandon documentation, contracts, processes, tools, and plans; rather, it places more emphasis on the people who are involved and on creating working software (Beck et al. 2001). Prior research identifies several benefits of agile methodologies in completing organizational projects including adaptability, flexibility, and project visibility (Kenefick 2011). Overviews of agile describe a number of methodologies that subscribe to the principles of the agile manifesto including Scrum and Kanban (Abrahammson et al. 2002; Wester 2014). Extant literature elaborates on the merit of these methodological frameworks for delivering project requirements using adaptive and agile processes (Baskerville et al. 2003, Daneva et al. 2013).

Although we recognize that the roots of agile project management stem from fields both inside and outside of the business literature (Sutherland and Schwaber 1995), this study examines the concept of agility within the software development context. We use the twelve principles behind the agile manifesto (Beck et al. 2001) along with extant literature (Conboy 2009, Lee and Xia 2010) to ground our notion of agile software development in order to gain an understanding of how agility is introduced and sustained in the software development practice. Thus, we define the practice of *agile software development* as a software team's ability to respond to changing user requirements through a process of continual readiness (Conboy 2009; Lee and Xia 2010). Through

agile practices, software teams may quickly and inherently create, embrace, and learn from change while contributing to perceived customer value (Conboy 2009).

APPENDIX A2

Key Terms:

- **Acceptance-routinization gap** can be defined as the difference between the patterns of cumulative assimilation events of an innovation across adopters at the acceptance stage and those at the routinization stage.
- **Agile acceptance** is the introductory adoption and employment of agile methods as a process innovation for one or more software development projects (Vijayasarathy and Turk 2012; Wang et al. 2012). We theorize about the use of ASD method, assuming the full use.
- **Agile methods** refer to process frameworks that are used by practitioners to develop software such as Scrum and eXtreme Programming.
- **Agile techniques** are sub-practices of agile methods
- **Agile routinization** is the usage of agile methods as a normal activity in the organization; the innovation is no longer considered out of the ordinary (Wang et al. 2012).
- **Agile software development (ASD)** can be defined as a software development team's ability to respond to changing user requirements through a process of continual readiness (Conboy 2009; Lee and Xia 2010).
- **Agile software development (ASD) assimilation** is the extent to which the use of ASD methods diffuses across the organizational projects or work processes and becomes routinized in the activities of those projects and processes (Cooper and Zmud 1990; Wang et al. 2012).
- **Assimilation gaps** can be defined as the difference between the patterns of cumulative assimilation events of an innovation across a population of adopters (Fichman and Kemerer 1999).
- **Competing Values Model**
  a. Group culture (change and internal focus) is primarily concerned with human relations and flexibility. Belonging, trust, and participation are its core values. Effectiveness criteria include the development of human potential and member commitment.
  b. Developmental culture (change and external focus) is future-oriented, considering what might be. The effectiveness criteria emphasize growth, resource acquisition, creativity and adaptation to the external environment.

c. Hierarchical culture (stability and internal focus) is oriented toward security, order, and routinization. It emphasizes control, stability and efficiency through the following of regulations.

d. Rational culture (stability and external focus) is achievement-oriented, focusing on productivity, efficiency, and goal achievement.

- **Diffusion of innovation theory** (DIO) refers to "the process by which an innovation is communicated through certain channels over time among the members of a social system" (Rogers 1962).

- **Innovation assimilation (IA)** is defined as the extent to which the use of an innovation diffuses across "the organizational projects or work processes and becomes routinized in the activities of those projects and processes" (Purvis, Sambamurthy, and Zmud 2001).

- **Organizational culture** is "a pattern of shared basic assumptions learned by an [organization] as it solves its problems of external adaptation and internal integration, which has worked well enough to be considered valid and, therefore, to new members as the correct way to perceive, think, and feel in relation to those problems" (Schein 1985).

- **Organizational structure** is "the logically consistent clustering of an organization's elements as it searches for harmony in its internal processes and consonance with its environment" (Mintzberg 1980).

- **Perceived ASD success** is the extent to which an organization's ASD project(s) meets technical goals, remains within the budget, is delivered in time, and is accepted by the end user (Jiang, Klein, and Pick 2003; Procaccino and Verner 2006).

- **Process innovation is** a process that is perceived as new by individuals or other units of adoption (Rogers 1983).

- **Six-stages of innovation assimilation:**

  Adoptive stages
  1. *Initiation*: need for change is recognized, a match is identified between an innovation and its application in the organization
  2. *Adoption*: a decision is made to adopt an innovation
  3. *Adaptation*: an adaptation to suit the contextual needs

  Post-adoptive stages
  4. *Acceptance*: use of the innovation
  5. *Routinization*:c an increase in the extent and intensity of use (i.e. usage of the innovation is encouraged as a normal activity)
  6. *Infusion*: increased usage in a more comprehensive and integrated manner results in increased effectiveness of systems development (i.e. the innovation penetrating deeply into an organization).

APPENDIX A3

Additional Details on Search Strategy

Our research protocol includes a detailed search strategy, inclusion and exclusion criteria, practical and detailed screening criteria, data extraction, and methods of synthesis. The majority of the protocol was used in the development of the reviewer training manual. Each step was detailed for replication in future studies, and by specifying the search criteria in the protocol beforehand, the reviewers were able to minimize the effects of a selection bias. The training manual was used to ensure procedural consistency in the execution of this study among its reviewers. Experts in the IS field were consulted concerning the search methodology so as to ensure completeness of the literature search before moving onto the analysis[4] (Petticrew & Roberts 2006, Fink 2005, Okoli 2010).

C.1 Citation Management

The reviewers managed 154 relevant citations using End Note X7 and End Note Web. The literature search results were imported into End Note, where the citations were sorted and folders were created in preparation for the inclusion and exclusion decisions made during both the practical and detailed screens. The practical screen was performed using the imported citation information for each article, including its title, keywords, abstract, and journal name, which were visible within the program.

---

[4] Experts included [expert 1] and [expert 2].

C.2 Reviewer Training

A training manual was developed so that each reviewer could be thoroughly trained in note taking and review techniques (Fink 2005; Ridley 2008; Okoli 2010). The reviewers used a computer-based note-taking strategy to extract data from eligible articles. Categories of study descriptives, theoretical descriptives, and agile life cycle descriptives were designated in order to identify and to describe theoretical perspectives within the articles.

Study descriptives consisted of each article's title, author(s), journal title, year of publication, research aim, theoretical lens, independent and dependent variables, sample, data collection, level of analysis, summary of the findings, and quality score. Theoretical descriptives included each article's ontological and epistemological assumptions, research approach, research method, and research techniques from Iivari et al's Information Systems Development framework (2004). Agile life cycle descriptives noted the development team's current stage in the software development life cycle during the study; agile adoption, agile routinization, agile infusion, or agile outcomes. These categories were adapted from previous IS studies that use assimilation stages based on (Cooper and Zmud 1990; Gallivan 2001; Wang et al. 2012).

A total of three reviewers examined each article included in this study, which consisted of two reviewers for selecting articles for inclusion and a third reviewer who acted as the "gold standard," or the deciding vote, if a disagreement occurred that would prevent the study from moving forward.

Reviewer Training Manual

Table of Contents

1. Introduction

1.1 Purpose and scope of this review: The purpose of this review is to assess the theoretical perspectives that influence our understanding of agile software development *within the IS field.*

1.2 Why this review is necessary? This review is necessary for the following reasons: (1) to develop a framework for a theoretical understanding of agile methods, which addresses the need for a unified framework for understanding agile expressed by Dingsøyr et al. (2012) and (2) to create an exemplar review containing the level of rigor suggested by The Campbell collaboration (campbellcollaboration.org), Arlene Fink (2005), Okoli (2010) and others. This review will detail each step so that the results can be replicated.

2. Applying eligibility criteria: The screening survey

2.1 Practical screen – Part 1: The following screen is designed to quickly identify articles that do not fit the basic qualification criteria for this study. This initial screen is intended to screen a number of agile studies without having to read beyond the abstract.

Instructions: Read and answer each question. If the answer to any of the following questions is **no**, the study is not eligible for this review and the corresponding article should be put into a folder prepare for articles that fail the practical screen.

Table C1. Practical Screen Eligibility Criteria
1. Is the study published in a journal that is listed on the Association for Information Systems' (AIS) webpage titled: MIS Journal Rankings (see Appendix A4)?   Yes………………Proceed to question #2   No…………………Stop. Place citation in folder labeled "Failed Practical Screen (Q1)"

2. Does the title, abstract, or keywords indicate that the study's main focus is on agile as a philosophy, approach, or methodology *of developing software*?
    a. The term 'agile' refers to a flexible and iterative method of developing software, which differs from traditional, plan-based approaches.
    b. Note: The term 'agile' should be used in relation to developing software as opposed to a form of project management for another activity, such as coal mining or manufacturing.
 Yes………………Study is eligible for the detailed screen.
 No……………..Stop. Place citation in folder labeled "Failed Practical Screen (Q2)"

End of Practical Screen

2.2 Detailed screen – Part 2: The following screen is designed to provide a detailed analysis of each article to determine whether it fits the eligibility criteria of this study. Area 1 of this screen should be applied to the abstract, theory development, and other relevant sections of articles that that have passed the practical screen. Area 2 of this screen should be applied to the research design, data collection, data analysis, and findings sections of articles that have passed the theoretical screen.

Instructions: Read and answer each of the questions for every available article. For questions in Area 1, if the answer to any of the following questions is no, then the study is not eligible for this review and the corresponding article should be put into a folder prepare for articles that fail the detailed screen. Only articles that have answered **yes to all** Area 1 questions are eligible for Area 2 question. For eligible

articles, the detailed screen should be performed in its entirety. Assign **one point** for

each **yes**. Articles must receive a score of 5 or more to be included in this review.

Table C2. Eligibility Criteria		
Areas	Questions	Answers
Area 1: Theoretical Content screen	1. Does the study examine agile software development holistically as opposed to a focus on a single technique or practice, such as user stories, paired programming, unit testing, and release planning? • Does the study intend to provide an understanding of agile software development methods? • Is the research focused on broad aspects of agile software development (adoption, implementation, management, etc…) rather than narrow details of agile techniques?	Yes or No
	2. Does the study present empirical data? • Is the study based on empirical data rather than conceptual or other types of non-empirical data? • Is the study based on scientific research rather than lessons learned, strategies, issues and challenges, or expert opinion?**	Yes or No
	3. Is a theory of interest or theoretical lens clearly stated? • Is there a clear mention of an existing or new theory? • Is there a theoretical basis for the constructs and proposed inter-relationships? • Is a theoretical lens or framework employed to understand agile software development? • Is the research focused on the underlying assumptions of agile software development	Yes or No
Area 2: Methodological Quality Screen	4. Is the research question clearly stated? • Is there a rationale for why the study was undertaken?** • Is there a clear statement of the study's primary outcome (i.e. time-to-market, cost, or product or process quality)?**	Yes or No

(Table C2 continued)

Table C2. Eligibility Criteria		
Areas	Questions	Answers
	5. Is there an adequate description of the context in which the research was carried out?** • The industry in which products are used (e.g. banking, telecommunications, consumer goods, travel, etc.)** • The nature of the software development organization (e.g. in-house department or independent software supplier)** • The experience of software development staff (e.g. with a language, a method, a tool, an application domain)**	Yes or No
	6. Is the research design specified? • Is the research design appropriate for answering the research question? • If an experiment was used, was there a control group with which to compare treatments?	Yes or No
	7. Are the measurement methods explicitly stated? • Are the operational definitions justified? • Are the constructs aggregated appropriately according to the level of analysis? • If the methods were modified during the study, has the researcher explained how and why? ** • Has the researcher justified the methods that were chosen? **	Yes or No
	8. Is the sampling method clearly stated? • Is the sample appropriate for generalizing to this population?	Yes or No
	9. Was the data collected in a way that addressed the research issue? • Is it clear how data was collected (e.g. semi-structured interviews, focus group etc.)? ** • Whether quality control methods were used to ensure completeness and accuracy of data collection**	Yes or No
	10. Was the data analysis sufficiently rigorous? • Was there an in-depth description of the analysis process? ** • If thematic analysis was used, is it clear how the categories/ themes	Yes or No

(Table C2 continued)

Table C2. Eligibility Criteria		
Areas	Questions	Answers
	were derived from the data? ** • Has sufficient data been presented to support the findings? ** • To what extent has contradictory data been taken into account? ** • Whether quality control methods were used to verify the results**	
	11. Is there a clear statement of findings? • Are the conclusions justified by the results? ** • Is the overall study believable?	Yes or No
	Additional comments:	
	**From Dybå and Dingsøyr (2008)	

3. Pilot Test of Review Process

1. Prior to applying the detailed screen to all the remaining studies, randomly select and apply the screen to only five eligible studies.

2. Revise the screening process as necessary to increase the accuracy and clarity of the instrument. 3. Once consistent results are achieved between reviewers, increase the pilot test to include 15-20 studies and compare results between raters.

4. Revise the screening process again if necessary until consistent results are achieved. Use the final version of the instrument to rate all the remaining studies.

5. If inter-rater agreement is at least 60 percent, discuss and reconcile differences. If agreement cannot be met between two reviewers, a third reviewer should act as the "gold standard" (Fink 2005) for resolving disagreements and keeping the process moving.

APPENDIX A4

AIS Journal Ranking

Table D1. AIS Journal Ranking List		
Rank	Journal Code	Journal Name
1	MISQ	Management Information Systems Quarterly
2	ISR	Information Systems Research
3	CACM	Communications of the ACM
4	MS	Management Science
5	JMIS	Journal of Management Information Systems
6	AI	Artificial Intelligence
7	DSI	Decision Sciences
8	HBR	Harvard Business Review
9	IEEETrans	Transactions
10	AIMag	AI Magazine
11	EJIS	European Journal of Information Systems
12	DSS	Decision Support Systems
13	IEEESw	Software
14	I&M	Information and Management
15	ACMTDS	ACM Transactions on Database Systems
16	IEEETSE	Transactions onSoftware Engineering
17	ACMTrans	ACM Transactions
18	JCSS	Journal of Computer and System Sciences
19	SMR	Sloan Management Review
20	CAIS	Communications of the AIS
21	IEEETSMC	IEEE Transactions on Systems, Man, and Cybernetics
22	ACS	ACM Computing Surveys
23	JComp	Journal on Computing
24	AMJ	Academy of Management Journal
25	IJEC	International Journal of Electronic Commerce
26	JAIS	Journal of AIS
27	IEEETC	Transactions on Computers
28	ISF	Information Systems Frontiers
29	JMS	Journal of Management Systems
30	OS	Organization Science
31	IEEEComp	Computer
32	ISJ	Information Systems Journal
33	ASQ	Administrative Science Quarterly

(Table D1 continued)

Rank	Journal Code	Journal Name
34	JGIM	Journal of Global Information Management
35	DATABASE	The DATABASE for Advances in Information Systems
36	JDM	Journal of Database Management
37	IS	Information Systems
38	MISQD	MIS Discovery
39	AMR	Academy of Management Review
40	JACM	Journal of ACM
41	COR	Computer and Operations Research
42	HCI	Human-computer interaction
43	CMR	California Management Review
44	IT&P	Information Technology & People
45	JSIS	Journal of Strategic Information Systems
46	JGITM	Journal of Global Information Technology Manangement
47	ACMTIS	ACM Transactions on Information Systems
48	InfoSci	Informing Science
49	JIM	Journal of Information Management
50	OR	Operations Research
51	JCIS	Journal of Computer Information Systems
52	BH	Business Horizons
53	IEEETKDE	IEEE Transactions on Knowledge and Data Engineering
54	JDA	Journal of Database Administration
55	IBMSJ	IBM Systems Journal
56	InfoSys	Infosystems
57	JITTA	Journalof Information Technology Theory and Application
58	KBS	Knowledge Based Systems
59	CompDcsn	Computer Decisions
60	IT&M	Information Technology and Management
61	WIRT	WIRT (Wirtschaftsinformatik
62	I&O	Information & Organization
63	ACMSIG	ACM Special Interest Group Publications
64	ESA	Expert Systems with Applications
65	ISM	Information Systems Management
66	INTFCS	Interfaces (INFORMS)
67	Omega	Omega
68	IJHCS	International Journal of Human-Computer Studies
69	DB	Database
70	JS&S	Journal of Systems and Software
71	DataMgmt	Data Management
72	IJMMS	International Journal of Man-Machine Studies
73	JIS(Acct)	Journal of Information Systems (accounting)

(Table D1 continued)

Rank	Journal Code	Journal Name
74	JISM	Journal of Information Systems Management
75	JIT	Journal of Information Technology
76	JOR	Journal of Operations Research
77	JOCEC	Journal of Organizational Computing
78	IRMJ	Information Resources Management Journal
79	JITCA	Journal of IT Cases and Application
80	JISE	Journal of Information Systems Education
81	JSM	Journal of Systems Management
82	JASIS	Journal of the American Society for Information Science
83	OBHDP	Organizational Behavior and Human
84	EMkt	Electronic Markets
85	AJIS	Australian Journal of Information Systems
86	JOEUC	Journal of Organizational & End User Computing
87	CSCW	Computer Supported Cooperative Work
88	JISci	Journal of Information Science
89	Dtmn	Datamation
90	INFOR	INFOR
91	IJIM	International Journal of Information Management
92	JITM	Journal of Information Technology Management
93	BIT	Behavior and Information Technology
94	ESR	Expert Systems Review
95	JEMIS	Journal of Education for Management Information Systems
96	CompJ	Computer Journal
97	IPM	Information Processing and Management
98	ECRA	Electronic Commerce Research and Application
99	IJTM	International Journal of Information Technology & Management
100	JIS(Edu)	Journal of Information Systems (education)
101	CHB	Computers in Human Behavior
102	EJOR	European Journal of Operations Research
103	TIS	The Information Society
104	CommRsch	Communication Research
105	IR	Information Research
106	JIIM	Journal of International Information Management
107	ESJ	E-Service Journal
108	IST	Information & Software Technology
109	Sim	Simulation
110	DPD	Database Programming and Design

Table E1. Articles that were eliminated during the screening process:	
Q1 – Published sources not found on AIS' journal list	
1. Dingsøyr and Hanssen	2002
2. Holz and Maurer	2002
3. Sena and Shan	2002
4. Erdogmus and Williams	2003
5. Zimmer	2003
6. Doran	2004
7. Fang et al.	2004
8. Koch	2004
9. Bellini et al.	2005
10. Hazzan and Dubinsky	2005
11. Crawford et al.	2006
12. Guntamukkala et al.	2006
13. LÓPez-Nores et al.	2006
14. Mason et al.	2006
15. Northover et al.	2006
16. Sfetsos et al.	2006
17. Turnu et al.	2006
18. Socha and Walter	2007
19. Ionel	2008
20. Layman et al.	2008
21. Mafakheri et al.	2008
22. Pikkarainen et al.**	2008
23. Salazar-Torres et al.	2008
24. Sharp and Robinson	2008
25. Cagley Jr.	2009
26. Ionel	2009
27. Johannessen and Ellingsen	2009
28. Levardy and Browning	2009
29. Sfetsos et al.	2009
30. Whelan	2009
31. Bonner	2010
32. Abdi and Labib	2011
33. Denning	2011
34. Diefenbach	2011
35. Ben-David et al.	2012
36. Israilidis and Jackson	2012
37. Lane and Gobet	2012
38. Mahnic	2012

(Table E1 continued)

Q1 – Published sources not found on AIS' journal list	
39. Nilsson and Wilson	2012
40. Overhage and Schlauderer**	2012
41. Pillai et al.	2012
42. Putnik and Putnik	2012
43. Skopik et al.	2012
44. Alaa and Fitzgerald	2013
45. Lu and Lu	2013
46. Birkinshaw	2014
47. Boschetti et al.	2014
48. Cervone	2014
49. Conforto et al.	2014
50. Senapathi and Srinivasan**	2014
Q2 – Not about agile software development	
1. Zhang et al.	2007
2. Joshi, Sarda, and Tripathi	2010
3. Kelly	2011
4. Allman	2012
5. Denning	2012
6. Jacobson et al.	2012
7. Wang, Conboy, and Cawley	2012
8. Keith, Demirkan, and Goul	2013
9. Pass and Ronen	2014
Q3 – Not a holistic perspective on agile	
1. Balijepally and Nerur	2006
2. Lee, DeLone, and Espinosa	2006
3. Simons	2006
4. Succi	2006
5. Wagstrom	2006
6. Arisholm et al.	2007
7. Xu and Ramesh	2007
8. Choi et al.	2008
9. Trinidad et al.	2008
10. Balijepally et al.	2009
11. Cummings, Espinosa, and Pickering	2009
12. Maruping, Zhang, and Venkatesh	2009
13. McAvoy and Butler	2009
14. Miranda, Bourque, and Abran	2009
15. Port and Bui	2009
16. Yadav et al.	2009
17. Falessi et al.	2010
18. Hannay et al.	2010
19. Tiwana	2010

(Table E1 continued)

Q3 – Not a holistic perspective on agile	
20. van Valkenhoef, Tervonen, and de Brock	2011
21. Warnars	2011
22. Bjarnason, Wnuk, and Regnell	2012
23. Ghanam, Maurer, and Abrahamsson	2012
24. Hanssen	2012
25. Mahnič and Hovelja	2012
26. Ramasubbu	2012
27. Daneva et al.	2013
28. Golfarelli et al.	2013
29. Hollis and Maiden	2013
30. Rafique and Misic	2013
31. Eck, Uebernickel, and Brenner	2014
32. Guerra	2014
33. Hansen and Lyytinen	2014
Q4 – Not an empirical study	
1. Dissanayake, Dantu, and Nerur	2013
2. Barlow et al.	2011
3. Harris, Hevner, and Collins	2009
4. Nerur and Balijepally	2007
5. Chan and Thong	2009
6. Scheerer	2014
7. Kakar	2014
8. Schmidt et al.	2013
9. Babb et al.	2014
10. Iivari and Iivari	2011
11. Yu and Petter	2014
12. Meso and Jain	2006
13. Austin and Devin	2009
14. Batra, VanderMeer, and Dutta	2011
15. Qumer and Henderson-Sellers	2008
16. Augustine et al.	2005
17. Turk, France, and Rumpe	2005
18. Vinekar et al.	2006
19. Dingsøyr et al.	2012
20. Cantor	2014
21. Senapathi, Drury, and Srinivasan	2013
Q5 – Lacks a theoretical assessment	
1. Chatterjee, Chakraborty, Sarker, and Sarker	2009
2. Laanti, Salo, and Abrahamsson	2011

3. Mishra, Mishra, and Ostrovska	2012
4. Fruhling and de Vreede	2006
5. Conboy	2009
Q6 – Lacks a methodological rigor	
1. Hadar, Sherman, and Hazzan	2008

** studies that were not found in the IS journal list, but pass the remaining screens

APPENDIX A6

Table F1. Theoretical Perspectives on Agile ISD					
Theory	Authors	Year	Type	Issue	IS/SE Search
1. Action Learning Theory	Nerur and Balijepally	2007	S	SP	IS
2. Adaptive Structuration Theory	Cao, Peng, and Ramesh**	2009	S	R	IS/SE
3. Agile Adoption and Improvement Model (AAIM)	Qumer and Henderson-Sellers	2008	S	R	IS
4. Agile Usage Research Model	Russo, Shams, and Fitzgerald	2013	S	R	IS
5. Collaborative learning	Hadar, Sherman, and Hazzan	2008	S	R	IS
6. Communication	Hummel and Rosenkranz	2014	S	R	IS
7. Competing Values Model of Organizational Culture	Iivari and Iivari	2011	S	R	IS
8. Complex Adaptive Theory	Socha and Walter	2007	S	R	SE
9. Complex Adaptive Theory	Levardy and Browning	2009	S	R	SE
10. Complex Adaptive Theory	Vidgen and Wang	2009	S	R	IS
11. Complex Adaptive Theory	Alaa and Fitzgerald	2013	S	R	IS
12. Complex Adaptive Theory	Meso and Jain	2006	S	R	SE
13. Complexity Theory	Falessi et al.	2010	S	R	SE
14. Contingency Theory	Austin and Devin	2009	S	R	IS
15. Contingency Theory	Sarker and Sarker	2009	M	R	IS
16. Control Theory	Harris, Hevner, and Collins	2009	S	SP	IS
17. Control Theory	Gregory, Sambhara, and Mathiassen	2013	S	R	IS
18. Control Theory	Persson, Mathiassen, and Aaen	2012	S	R	IS
19. Control Theory	Maruping, Viswanath, and Agarwal	2009	S	R	IS
20. Control Theory	Yadav et al.	2009	S	R	IS
21. Control Theory/ Dynamic Capabilities	Harris, Collins, and Hevner*	2009	S	SP	IS
22. Coordination Theory	Scheerer	2014	S	R	IS
23. Coordination Theory	Strode, Huff, Hope, and Link	2012	S	R	IS
24. Coordination Theory	Pikkarainen, Haikara, and Salo	2008	S	R	SE
25. Creativity	Dissanayake, Dantu, and Nerur	2014	S	R	IS
26. Creativity	Hollis and Maiden	2013	S	R	IS
27. Decision Making	Drury, Conboy, and Power	2012	S	R	IS
28. Definition of,	Conboy	2009	M	SP	IS
29. Definition of,	Baskerville, Pries-Heje, and Madsen	2011	M	SP	IS
30. Dialectic Theory	Vijayasarathy and Turk	2012	S	R	IS
31. Diffusion of Innovation (DOI) Theory	Senapathi and Srinivasan	2012	S	SP	IS
32. Diffusion of Innovation (DOI) Theory	Senapathi, Drury, and Srinivasan	2013	S	R	IS
33. Diffusion of Innovation (DOI) Theory	Senapathi and Srinivasan	2014	S	R	SE

Table F1. Theoretical Perspectives on Agile ISD					
34. Diffusion of Innovation (DOI) Theory	Hong, Thong, and Chasalow	2011	S	SP	IS
35. Diffusion of Innovation (DOI) Theory (Acceptance)	Mangalaraj, Mahapatra, and Nerur	2009	M	SP	IS
36. Diffusion of Innovation (DOI) Theory (Acceptance)	Overhage and Schlauderer	2012	M	R	IS
37. Diffusion of Innovation (DOI) Theory (Innovation Assimilation)	Eck, Uebernickel, and Brenner	2014	S	R	IS
38. Diffusion of Innovation (DOI) Theory (Innovation Assimilation)	Conboy, Pikkarainen, and Wang	2007	S	R	IS
39. Diffusion of Innovation (DOI) Theory (Innovation Assimilation)	Wang, Pikkarainen, and Conboy	2012	M	SP	IS
40. Distributed Cognition	Ramasubbu and Kemerer	2012	S	R	SE
41. Distributed Cognition	Hansen and Lyytinen	2014	S	R	IS
42. Distributed Cognition	Sharp and Robinson	2008	S	R	SE
43. Double loop learning	McAvoy and Butler*, **	2007	S	R	IS/SE
44. Dynamic Capabilities/Control theory	Harris, Collins, and Hevner*	2009	S	SP	IS
45. Evolutionary theory of knowledge	Northover, Boake, and Kourie	2006	S	R	SE
46. Expertise Coordination	Maruping, Zhang, Venkatesh	2009	S	R	IS
47. Game theory	Hazzan and Dubinsky	2005	S	R	SE
48. Home ground theory	Port and Bui	2009	S	R	IS
49. Input-Process-Output	Melo, Cruzes, Kon, and Conradi	2013	M	R	IS
50. Input-Process-Output	Wood, Michaelides, and Thomson	2013	M	R	IS
51. Job characteristics theory	Kakar	2014	S	R	IS
52. Knowledge management	Chan and Thong **	2009	S	R	IS/SE
53. Knowledge management	Holz and Maurer *	2002	S	R	SE
54. Knowledge management	Sena and Shan	2002	S	R	SE
55. Knowledge management	Doran	2004	S	R	SE
56. Knowledge management	Crawford et al.	2006	S	R	SE
57. Knowledge management	Salazar-Torres et al.	2008	S	R	SE
58. Knowledge management	Bellini et al.	2005	S	R	SE
59. Knowledge management	Dingsøyr and Hanssen **	2002	S	R	IS/SE
60. Knowledge Sharing	Santos et al.	2014	S	R	IS
61. Langrangian heuristic	Boschetti et al.	2014	S	R	IS
62. Materiality	Wagner et al.	2013	S	R	IS
63. Method for method config	Karlsson and Ågerfalk	2009	M	R	IS
64. Mindfulness	McAvoy, Nagle, and Sammon	2013	M	SP	IS
65. Organizational learning	Holz and Maurer*	2002	S	R	SE
66. Overview of Agile Principles in Larger, Dynamic Software Projects: AST, CAS, control, TCE, social exchange, expectancy	Batra, VanderMeer, and Dutta	2011	S	R	IS
67. Agile Implementation	Barlow et al.	2011	S	SP	IS
68. Personality	Balijepally and Nerur	2006	S	R	IS
69. Personality	Bishop and Deokar	2014	S	R	IS
70. Personality	Layman et al.	2008	S	R	SE
71. Personality	Sfetsos et al.	2009	S	R	SE
72. Personality	Choi et al.	2008	S	R	SE
73. Personality	Sfetsos et al.	2006	S	R	SE
74. Personality	Acuna et al.	2009	S	R	SE
75. Personality	Hannay et al.	2010	S	R	SE
76. REALM	Babb et al.	2014	S	R	IS

Table F1. Theoretical Perspectives on Agile ISD						
77.	Requirements Prioritization	Daneva et al.	2013	S	R	IS
78.	Self-organization	Hoda, Noble, and Marshall	2011	S	R	IS
79.	Self-organization	Hoda, Noble, and Marshall	2013	S	SP	IS
80.	Shared mental models	Schmidt, Kude, Heinzl, and Mithas	2014	S	R	IS
81.	Shared mental models	Ryan and O'Connor	2013	S	R	SE
82.	Shared mental models	Adolph, Kruchten, and Hall	2012	S	R	IS
83.	Shared Mental Models	Yu and Petter	2014	S	R	IS
84.	Social facilitation	Arisholm et al.	2007	S	R	SE
85.	Social facilitation	Balijepally et al.	2009	S	R	SE
86.	Social identity theory /self-categorization theory	Lee and Xia	2010	S	R	IS
87.	Social theory*	Drechsler and Trepper	2014	S	R	IS
88.	Team adaption theory	Schmidt, Kude, Tripp, Heinzl, and Spohrer	2013	S	R	IS
89.	Teamwork model	Moe et al.	2010	S	R	SE
90.	Theory of diagnosis	van Valkenhoef, Tervonen, and de Brock	2011	S	R	SE
91.	Theory of diagnosis	Trinidad et al.	2008	S	R	SE
92.	Triple-loop learning	McAvoy and Butler*, **	2007	S	R	IS/SE
93.	Trust	Goh, Pan, and Zuo	2013	M	R	IS
* = Duplicate entry; study contains multiple theoretical perspectives						
** = Study found using both IS and SE searches						

APPENDIX A7

Detailed Screen Descriptives

Table G1. Data Extraction Form
Study descriptives
Authors
Paper title (Title)
Journal title (Journal)
Year of publication (Year)
Purpose
Existing theory or theoretical lens
Main effects
Independent variables
Dependent variables
Sampling

Data collection
Level of Analysis
Summary of the Findings
Quality Score
Theoretical Descriptives
Paradigms
Ontology
Epistemology
Approach
Method
Technique
Agile Assimilation Stage Descriptives
Stage

Table G1 shows the fields that were collected in the data extraction form. The theoretical descriptives were constructed using Iivari et al.'s (2004) Information Systems Development (ISD) framework (shown in Figure G1). The ISD hierarchy represents the underlying philosophical assumptions that influenced the production of their research. This framework was integrated with the agile assimilation stage characteristic from the data extraction form in order to show the extent of our theoretical understanding underneath each agile adoption stage.

Four-Tiered Research Model

- Paradigms (underlying assumptions guiding research)
    o Ontology
    o Epistemology
    o Methodology
    o Ex. Functionalism and Interpretivism
- Approaches (a way of going about research)
    o Goals
    o Guiding Principles
    o Fundamental Concepts
    o Principles of the ISD Process
    o Ex. Conceptual and Historical
- Methods (procedures for conducting research)
    o Relationship Between
    o Techniques
    o Detailed ISD Process
    o Ex. Field research and Case Study
- Techniques (tools used in research)
    o Detailed Concepts
    o Notations
    o Ex. SEM and Nivo

Figure G1. Hierarchy of ISD paradigms, approaches, methods and techniques (Iivari et al. 2004)

Although most of the data of interest were clearly stated, some data points, particularly the theoretical descriptives and level of analysis, were not always explicitly stated. Only three of the final studies clearly stated the philosophical assumptions (i.e. *interpretive, positivist, or critical realist* case studies) that they used to theoretically ground their studies. In order to determine these assumptions, I examined the work of Devers (1999), Lincoln and Guba (1985), Orlikowski and Baroudi (1991), and Dube and Pare (2003) for criteria on how to properly classify theoretically grounded studies according to their philosophical assumptions.

The level of analysis, which refers to the major unit being analyzed in scientific research (Trochim 2006), was determined based on a careful inspection of the conceptualization of

the research questions, data collection instrument, data analysis process, and concluding inferences. For example, a study that aimed to increase the understanding of agile development both by surveying team members with questions pertaining to agile team success factors, and by using the data analysis to make inferences about agile teams would be classified as having a unit of analysis at the team level. A description of each level is available in Appendix A8. Overall, 23 of the final 37 studies clearly state the level of analysis, and the rest were deducted from the details listed above.

The researcher created the agile assimilation stage characteristic in order to locate the development team's current stage in the agile adoption process during the study. The agile adoption stage characteristic was determined from the study's motivation, research design, and project description. For example, if the study's aim was to understand adoption challenges facing agile teams, then the assimilation stage was placed under the Introduction and Adoption of Agile Methods column. Stages were classified as follows: Introduction and Adoption of Agile Methods column (adoption), which classifies software teams that have newly adopted agile methods; Acceptance/use of Agile Methods, which classifies software teams that have moved beyond the decision to adopt agile, but have not made agile ISD a regular part of their ISD process; Routine and Infusion Use of Agile Development Processes (routinization), which classifies software teams that have made agile methods a routine part of their development processes, demonstrate deep use of agile methods, and/or have relatively high amounts of experience using agile methods (see Appendix A8). Thus, we took both the depth of use and length of experience into account when classifying agile teams. Because no standard time ranges exist for determining a software team's level of maturity with agile methods,

the researcher used the following markers as a guide for separating highly experienced teams from less experienced teams: Introduction and Adoption of Agile Methods, 0-12 months; Acceptance/use of Agile Methods, 1-3 years; and Routine and Infusion Use of Agile Development Processes, 3 or more years.

APPENDIX A8

TPA Framework

We developed the Theoretical Perspectives on Agile Software Development Framework (TPA) to answer RQ3, "*Where are the gaps in our theoretical understanding of agile software development research?*" and to provide further insight into the research gaps from an assimilation perspective (see Figure H1).

The TPA framework organizes studies according to their stage of assimilation and level of analysis. Table H1 provides a description of the main headers of the TPA framework. The development and application of our TPA framework reveals significant gaps in the theoretical assessments in agile research such as the dearth of theoretical assessments at the routinization and infusion stages. We analyzed the 38 articles that passed the screening process using a more detailed data extraction form than the one applied to articles that only passed the practical screen.

Agile Assimilation Stages		
A	B	C
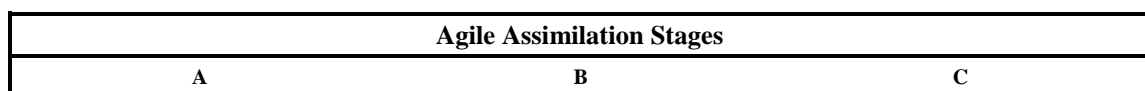
(Figure H1 continued)

UNIT	Introduction and Adoption of Agile Methods[1]	Acceptance/use of Agile Methods	Routine and Infusion Use of Agile Development Processes
	Less experienced		More experienced
ORG		Control theory (Gregory, Sambhara, and Mathiassen 2013) Knowledge Sharing (Santos et al. 2014) Shared mental models (Adolph et al. 2012)	Diffusion of Innovation (Senapathi and Srinivasan 2012; 2014) Socio-materiality (Wagner et al. 2013)
PROJ	Adaptive Structuration Theory (Cao et al. 2009)	Diffusion of Innovation (Acceptance) (Mangalaraj, Mahapatra, and Nerur 2009) Control/Dynamic Capabilities Theory (Harris et al. 2009) Coordination Theory (Strode et al. 2012) ***GSD*** Trust (Goh et al. 2013) Communication (Hummel, Rosenkranz, and Holten 2014)	
TEAM (3)	Double Loop/ Triple Loop Learning (McAvoy and Butler 2007) Coordination Theory (Pikkarainen et al. 2008) Method for Method Configuration (Karlsson and Ågerfalk 2009) Contingency Theory (Sarker et al. 2009) Teamwork (Moe et al. 2010) Diffusion of Innovation (Russo, Shams, and Fitzgerald 2013)	Diffusion of Innovation (Conboy, Pikkarainen, and Wang 2007) Control Theory (Maruping et al. 2009) (Persson et al. 2012) ***GSD*** Personality (Acuna et al. 2009) Complex Adaptive Theory (CAS) (Vidgen and Wang 2009) Social Identity Theory (Lee and Xia 2010) Social technical systems (Hoda, Noble, and Marshall 2011, 2013) Innovation Assimilation (Wang, Pikkarainen, and Conboy 2012) Mindfulness (McAvoy et al. 2013) Knowledge Transference (Ryan and O'Connor 2013) Shared Mental Models (Schmidt et al. 2014)	Input-Process-Output (Melo et al. 2013) (Wood et al. 2013)
IND	Dialectic Theory (Vijayasarathy and Turk 2012)	Diffusion of Innovation (Acceptance)	

(Figure H1 continued)

		(Hong et al. 2011)
	Descriptive Decision Making (Drury et al. 2012)	Diffusion of Innovation (Overhage and Schlauderer 2012)
		Personality (Bishop and Deokar 2014)

*** Denotes a Global Software Development (GSD) team
****Baskerville et al. (2011) encompasses all levels

Figure H1. Theoretical Perspectives in Agile Software Development Framework

Table H1. TPA Definitions

Levels of Analysis		Agile Assimilation Stage	
Level	Description	Stage	Description
Organization	The major unit being analyzed is the organization.	Introduction and Adoption of Agile Methods	This stage classifies software teams that teams that made the decision to adopt agile methods.
Project	The major unit being analyzed is the individual software project.	Acceptance/use of Agile Methods	This stage classifies software teams that the used agile methods in a general sense.
Team	The major unit being analyzed is the software team.	Routine and Infusion Use of Agile Development Processes	This stage classifies software teams that increased the extent and intensity of their agile use.
Individual	The major unit being analyzed is the individual developer.		

APPENDIX A9

Papers collected in the literature review with theoretical perspectives

Acuña, S. T., Gómez, M., and Juristo, N. 2009. "How do personality, team processes and task characteristics relate to job satisfaction and software quality?," *Information and Software Technology* (51:3), pp 627-639.

Adolph, S., Kruchten, P., and Hall, W. 2012. "Reconciling perspectives: A grounded theory of how people manage the process of software development," *Journal of Systems and Software* (85:6), pp. 1269-1286.

Alaa, G., and Fitzgerald, G. 2013. "Re-conceptualizing agile information systems development using complex adaptive systems theory," *Emergence: Complexity & Organization* (15:3).

Arisholm, E., Gallis, H., Dyba, T., and Sjoberg, D.I.K., 2007. "Evaluating pair programming with respect to system complexity and programmer expertise," *IEEE Transactions on Software Engineering* (33:2), pp. 65–86.

Austin, R., and Devin, L. 2009. "Weighing the Benefits and Costs of Flexibility in Making Software: Toward a Contingency Theory of the Determinants of Development Process Design," *Information Systems Research* (20:3), pp. 462-477.

Babb, J., Hoda, R., and Norbjerg, J. 2014. "Embedding Reflection and Learning into Agile Software Development," *IEEE software* (31:4), pp. 51-57.

Balijepally, V., Mahapatra, R., and Nerur, S. 2006. "Assessing Personality Profiles of Software Developers in Agile Development Teams," *Communications of AIS* (18:4), pp. 2-40.

Balijepally, V., Mahapatra, R., Nerur, S., and Price, K. H. 2009. "Are Two Heads Better Than One for Software Development? The Productivity Paradox of Pair Programming," *MIS Quarterly* (33:1), pp. 91-118.

Barlow, J. B., Giboney, J. S., Keil, M. J., Wilson, D. W., Schuetzler, R. M., Lowry, P. B., and Vance, A. 2011. "Overview and Guidance on Agile Development in Large Organizations," *Communications of AIS* (29:2), pp. 25-44.

Baskerville, R., Pries-Heje, J., and Madsen, S. 2011. "Post-agility: What follows a decade of agility?," *Information and Software Technology* (53:5), pp 543-555.

Batra, D., VanderMeer, D., Kaushik, D. 2011. "Extending Agile Principles to Larger, Dynamic Software Projects: A Theoretical Assessment," *Journal of Database Management* (22:4), pp. 73-92.

Bellini, E., Canfora, G., Garcia, F., Piattini, M., and Visaggio, C. A. 2005. "Pair designing as practice for enforcing and diffusing design knowledge," *Journal of Software Maintenance and Evolution-Research and Practice* (17:6) Nov-Dec, pp. 401-423.

Bishop, D., and Deokar, A. 2014. "Toward an Understanding of Preference for Agile Software Development Methods from a Personality Theory Perspective," *System Sciences (HICSS), 2014 47th Hawaii International Conference on,* IEEE, pp. 4749-4758.

Boschetti, M. A., Golfarelli, M., Rizzi, S., and Turricchia, E. 2014. "A Lagrangian heuristic for sprint planning in agile software development," *Computers & Operations Research* (43), pp. 116-128.

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

Chan, F. K., and Thong, J. Y. 2009. "Acceptance of agile methodologies: A critical review and conceptual framework," *Decision Support Systems* (46:4), pp. 803-814.

Choi, K.S., Deek, F.P., and Im, I., 2008. "Exploring the underlying aspects of pair programming: the impact of personality," *Information and Software Technology* (50), pp. 1114–1126.

Conboy, K. 2009. "Agility from first principles: reconstructing the concept of agility in information systems development," *Information Systems Research* (20:3), pp. 329-354.

Conboy, K., Pikkarainen, M., and Wang, X. 2007. "Agile practices in use from an innovation assimilation perspective: a multiple case study," *ICIS 2007 Proceedings*, p 7.

Crawford, B., Castro, C., and Monfroy, E. 2006. "Knowledge management in different software development approaches," in *Advances in Information Systems, Proceedings,* T. Yakhno and E. J. Neuhold (eds.), pp. 304-313.

Daneva, M., van der Veen, E., Amrit, C., Ghaisas, S., Sikkel, K., Kumar, R., Ajmeri, N., Ramteerthkar, U., and Wieringa, R. 2013. "Agile requirements prioritization in large-scale outsourced system projects: An empirical study," *Journal of Systems and Software* (86:5), pp. 1333-1353.

Dingsoyr, T., and Hanssen, G. K. 2002. "Extending agile methods: Postmortem reviews as extended feedback," in *Advances in Learning Software Organizations,* S. Henninger and F. Maurer (eds.), pp. 4-12.

Dissanayake, I., Dantu, R., and Nerur, S. 2013. "Knowledge Management in Software Development," in *Proceedings of the 19th Americas Conference on Information Systems,* Chicago, Illinois, December 10-13, pp. 1-4.

Doran, H. D. 2004. "Agile Knowledge Management in Practice," in *Advances in Learning Software Organizations, Proceedings,* G. Melnik and H. Holz (eds.), pp. 137-143.

Drechsler, A. and Trepper, T., 2014. "The Potential of Niklas Luhmann's General Theory of Social Systems for Research on Agile Methodologies," in *AMCIS.*

Drury, M., Conboy, K., and Power, K. 2012. "Obstacles to decision making in Agile software development teams," *Journal of Systems and Software* (85:6), pp 1239-1254.

Eck, A., Uebernickel, F., and Brenner, W. 2014. "Fit for Continuous Integration: How Organizations Assimilate an Agile Practice," *Proceedings of the 20th Americas Conference on Information Systems,* Savannah, Georgia, August 7-9, pp. 1-11.

Falessi, D., Cantone, G., Sarcia, S. A., Calavaro, G., Subiaco, P., and D'Amore, C. 2010. "Peaceful Coexistence: Agile Developer Perspectives on Software Architecture," *IEEE Software*, (27:2), pp. 23-25.

Fang, M., Ying, J., and Wu, M. H. 2004. "Effective Elements of Integrated Software Development Process Supported Platform," in *Computer Supported Cooperative Work in Design I,* W. Shen, Z. Lin, J. P. A. Barthes and T. Li (eds.), pp. 368-377.

Goh, J. C.-L., Pan, S. L., and Zuo, M. 2013. "Developing the Agile IS Development Practices in Large-scale IT Projects: The trust-mediated organizational controls and IT project team capabilities perspectives," *Journal of the Association for Information Systems* (14:12), pp. 722-756.

Gregory, T., Mathiassen, L., and Sambhara, C. 2013. "Chains of Control in Agile Software Development," in *Proceedings of the 19th Americas Conference on Information Systems*, Chicago, Illinois, August 15-17, pp. 1-9.

Hadar, I., Sherman, S., and Hazzan, O. 2008. "Learning Human Aspects of Collaborative Software Development," *Journal of Information Systems Education* (19:3) Fall2008, pp. 311-319.

Hannay, J.E., Arisholm, E., Engvik, H., and Sjoberg, D.I.K., 2010. "Effects of personality on pair programming," *IEEE Transactions on Software Engineering*, (36:1), pp. 61–80.

Hansen, S., and Lyytinen, K. 2014. "Requirements Computation: Analyzing Requirements Cognition in Multiple Development Paradigms," in *Proceedings of the 20th Americas Conference on Information Systems, Savannah, Georgia*, pp. 1-13.

Hanssen, G. K. 2012. "A Longitudinal Case Study of an Emerging Software Ecosystem: Implications for Practice and Theory," *Journal of Systems & Software* (85:7), pp. 1455-1466.

Harris, M. L. Hevner, A. R., and Collins, R. W. 2009. "Controls in Flexible Software Development," *Communications of AIS* (24), pp. 757-776.

Harris, M. L., Collins, R. W., and Hevner, A. R. 2009. "Control of Flexible Software Development Under Uncertainty," *Information Systems Research* (20:3), pp. 400-419.

Hazzan, O., and Dubinsky, Y. 2005. "Social Perspective of Software Development Methods: The Case of the Prisoner Dilemma and Extreme Programming," in *Extreme Programming and Agile Processes in Software Engineering, Proceedings,* H. Baumeister, M. Marchesi and M. Holcombe (eds.), pp. 74-81.

Hoda, R., Noble, J., and Marshall, S. 2011. "The Impact of Inadequate Customer Collaboration on Self-organizing Agile Teams," *Information and Software Technology* (53:5), pp. 521-534.

Hoda, R., Noble, J., and Marshall, S. 2013. "Self-Organizing Roles on Agile Software Development Teams," *IEEE Transactions on Software Engineering* (39:3), pp. 422-444.

Hollis, B., and Maiden, N. 2013. "Extending Agile Processes with Creativity Techniques," *Software, IEEE* (30:5), pp. 78-84.

Holz, H., and Maurer, F. 2002. "Knowledge management support for distributed agile software processes," in *Advances in Learning Software Organizations,* S. Henninger and F. Maurer (eds.), pp. 60-80.

Hong, W., Thong, J. Y., Chasalow, L. C., & Dhillon, G. 2011. "User acceptance of agile information systems: A model and empirical test," *Journal of Management Information Systems* (28:1), pp 235-272.

Hummel, M., Rosenkranz, C., and Holten, R. 2013. "Explaining The Changing Communication Paradigm Of Agile Information Systems Development: A Research Model, Measurement Development And Pretest," in *Proceedings of the 21st European Conference on Information Systems*, p. 17.

Iivari, J. and Iivari, N. 2011. "The Relationship Between Organizational Culture and the Deployment of Agile Methods," *Information and Software Technology* (53:5), pp 509-520.

Johannessen, L.K., and Ellingsen, G., 2009. "Integration and Generification-agile Software Development in the Healthcare Market," *Computer Supported Cooperative Workthe Journal of Collaborative Computing*, (18:5), pp. 607–634.

Kakar, A. K. 2014. "Teaching Theories Underlying Agile Methods in a Systems Development Course," *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, IEEE, pp. 4970-4978.

Karlsson, F. Å., and Pär J 2009. "Towards structured flexibility in information systems development: Devising a method for method configuration," *Journal of Database Management (JDM)* (20:3), pp. 51-75.

Keith, M., Demirkan, H., and Goul, M. 2013. "Service-oriented Methodology for Systems Development," *Journal of Management Information Systems* (30:1), pp. 227-260.

Layman, L., Williams, L., Slaten, K., Berenson, S., and Vouk, M., 2008. "Addressing Diverse Needs Through a Balance of Agile and Plan-Driven Software Development Methodologies in the Core Software Engineering Course," *International Journal of Engineering Education,* (24), pp. 659–670.

Lee, G., and Xia, W. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data," *MIS Quarterly* (34:1), pp 87-114.

Lévárdy, V., and Browning, T. R. 2009. "An Adaptive Process Model to Support Product Development Project Management," *IEEE Transactions on Engineering Management,* (56:4), pp. 600–620.

Mafakheri, F., Nasiri, F., and Mousavi, M. 2008. "Project agility assessment: an integrated decision analysis approach," *Production Planning & Control* (19:6), pp. 567-576.

Mangalaraj, G., Mahapatra, R., and Nerur, S. 2009. "Acceptance of software process innovations–the case of extreme programming," *European Journal of Information Systems* (18:4), pp. 344-354.

Maruping, L. M., Venkatesh, V., and Agarwal, R. 2009. "A Control Theory Perspective on Agile Methodology Use and Changing User Requirements," *Information Systems Research* (20:3), pp. 377-399.

Maruping, L. M., Zhang, X., and Venkatesh, V. 2009. "Role of Collective Ownership and Coding Standards in Coordinating Expertise in Software Project Teams," *European Journal of Information Systems* (18:4), pp 355-371.

McAvoy, J. and Butler, T. 2009. "The Role of Project Management in Ineffective Decision Making within Agile Software Development Projects," *European Journal of Information Systems*, (18:4), pp. 372-383

McAvoy, J., and Butler, T. 2007. "The impact of the Abilene Paradox on double-loop learning in an agile team," *Information and software technology* (49:6), pp. 552-563.

McAvoy, J., Nagle, T., and Sammon, D. 2013. "Using mindfulness to examine ISD agility," *Information Systems Journal* (23:2), pp. 155-172.

Melo, C. d. O., Cruzes, D. S., Kon, F., and Conradi, R. 2013. "Interpretative case studies on agile team productivity and management," *Information and Software Technology* (55:2), pp 412-427.

Meso, P., and Jain, R. 2006. "Agile Software Development: Adaptive Systems Principles and Best Practices," *Information Systems Management* (23:3) Summer2006, pp. 19-30.

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* (52:5), pp. 480-491.

Nerur, S., and Balijepally, V.G. 2007. "Theoretical Reflections on Agile Development Methodologies," *Communications of the ACM* (50:3), pp. 79-83.

Northover, M., Boake, A., and Kourie, D. G. 2006. "Karl Popper's critical rationalism in agile software development," in *Conceptual Structures: Inspiration and Application,* H. Scharfe, P. Hitzler and P. Ohrstrom (eds.), pp. 360-373.

Overhage, S., and Schlauderer, S. 2012. "Investigating the long-term acceptance of agile methodologies: An empirical study of developer perceptions in scrum projects," in the *45th Hawaii International Conference on System Sciences (HICSS),* pp. 5452-5461.

Persson, J. S., Mathiassen, L., and Aaen Ivan 2012. "Agile distributed software development: enacting control through media and context," *Information Systems Journal* (22:6), pp. 411-433.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., and Still, J. 2008. "The impact of agile practices on communication in software development," *Empirical Software Engineering* (13:3), pp. 303-337.

Port, D., and Bui, T. 2009. "Simulating Mixed Agile and Plan-based Requirements Prioritization Strategies: Proof-of-concept and Practical Implications," *European Journal of Information Systems* (18:4), pp. 317-331.

Qumer, A., and Henderson-Sellers, B. 2008. "A framework to support the evaluation, adoption and improvement of agile methods in practice," *Journal of Systems & Software* (81:11), pp. 1899-1919.

Ramasubbu, N., Kemerer, C. F., and Hong, J. 2012. "Structural complexity and programmer team strategy: An experimental test," *Software Engineering, IEEE Transactions on* (38:5), pp. 1054-1068.

Russo, N. L., Fitzgerald, G., and Shams, S. 2013. "Exploring Adoption and Use of Agile Methods: A Comparative Case Study," in *Proceedings of the 19th Americas Conference on Information Systems*, Chicago, Illinois, August 15-17, pp. 1-8.

Ryan, S., and O'Connor, R. V. 2013. "Acquiring and sharing tacit knowledge in software development teams: An empirical study," *Information and Software Technology* (55:9), pp 1614-1624.

Salazar-Torres, G., Colombo, E., Da Silva, F.S.C., Noriega, C.A., and Bandini, S., 2008. "Design issues for knowledge artifacts," *Knowledge-Based Systems*, (21), pp. 856–867.

Santos, V., Goldman, A., Martins, D., and Cortes, M. 2014. "The Influence of Organizational Factors on Inter-team Knowledge Sharing Effectiveness in Agile Environments," *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, IEEE, pp. 4729-4738.

Sarker, S., and Sarker, S. 2009. "Exploring agility in distributed information systems development teams: an interpretive study in an offshoring context," *Information Systems Research* (20:3), pp 440-461.

Scheerer, A., Hildenbrand, T., and Kude, T. 2014. "Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective," *System Sciences (HICSS), 2014 47th Hawaii International Conference on, IEEE*, pp. 4780-4788.

Schmidt, C., Kude, T., Heinzl, A., and Mithas, S. 2014. "How Agile Practices Influence the Performance of Software Development Teams: The Role of Shared Mental Models and Backup," in *Proceedings of the 35th International Conference on Information Systems, Auckland*, New Zealand, pp. 1-18.

Schmidt, C., Kude, T., Tripp, J., Heinzl, A., and Spohrer, K. 2013. "Team Adaptability in Agile Information Systems Development," in *Proceedings of the 34th International Conference on Information Systems*, Milan, Italy, pp. 1-11.

Sena, J. A., and Shan, A. B. 2002. "Integrating knowledge management, learning mechanisms, and company performance," in *Practical Aspects of Knowledge Management,* D. Karagiannis and U. Reimer (eds.), pp. 620-631.

Senapathi, M., and Srinivasan, A 2012. "Understanding post-adoptive agile usage: An exploratory cross-case analysis," *Journal of Systems and Software* (85:6), pp 1255-1268.

Senapathi, M., and Srinivasan, A. 2014. "An Empirical Investigation of the Factors Affecting Agile Usage," *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ACM, p. 10.

Senapathi, M., Drury, M., and Srinivasan, A. 2013. "Agile Usage: Refining a Theoretical Model," in *Proceedings of the Pacific Asia Conference Information Systems*, p. 43.

Sfetsos, P., Stamelos, I., Angelis, L., and Deligiannis, I., 2006. "Investigating the Impact of Personality Types on Communication and Collaboration-viability in Pair Programming—an Empirical Study," in *Extreme Programming and Agile Processes in Software Engineering, Proceedings* Abrahamsson, P., Marchesi, M., Succi, G. (Eds.), Springer Berlin Heidelberg, pp. 43–52.

Sfetsos, P., Stamelos, I., Angelis, L., and Deligiannis, I., 2009. "An Experimental Investigation of Personality Types Impact on Pair Effectiveness in Pair programming," *Empirical Software Engineering* (14:2), pp. 187–226.

Sharp, H. and Robinson, H. 2008. "Collaboration and co-ordination in mature eXtreme programming teams," *International Journal of Human-Computer Studies*, (66:7), pp. 506–518.

Socha, D., and Walter, S. 2007. "Is designing software different from designing other things?" *International Journal of Engineering Education* (22:3), p 540.

Strode, D. E., Huff, S. L., Hope, B., and Link, S. 2012. "Coordination in co-located agile software development projects," *Journal of Systems and Software* (85:6), pp 1222-1238.

Tiwana, A. 2010. "Systems development ambidexterity: Explaining the complementary and substitutive roles of formal and informal controls," *Journal of Management Information Systems* (27:2), pp. 87-126.

Trinidad, P., Benavides, D., Duran, A., Ruiz- Cortés, A., and Toro, M. 2008. "Automated Error Analysis for the Agilization of Feature Modeling," *Journal of Systems and Software*, (81:6), pp. 883–896.

Vidgen, R., and Wang, X. 2009. "Coevolving systems and the organization of agile software development," *Information Systems Research* (20:3), pp. 355-376.

Vijayasarathy, L., and Turk, D. 2012. "Drivers of agile software development use: Dialectic interplay between benefits and hindrances," *Information and Software Technology* (54:2), pp 137-148.

Wagner, E. L., Newell, S., and Ramiller, N. 2013. "The Structuring of Freedom in Agile Development," in *Proceedings of the 19th Americas Conference on Information Systems*, Chicago, Illinois, August 15-17, pp. 1-7.

Wang, X., Conboy, K., and Pikkarainen, M. 2012. "Assimilation of agile practices in use," *Information Systems Journal* (22:6), pp. 435-455.

Wood, S., Michaelides, G., and Thomson, C. 2013. "Successful extreme programming: Fidelity to the methodology or good teamworking?," *Information and Software Technology* (55:4), pp. 660-672.

Xu, P., and Ramesh, B. 2007. "Software Process Tailoring: An Empirical Investigation," *Journal of Management Information Systems* (24:2), pp. 293-328.

Yadav, V., Adya, M., Sridhar, V., and Nath, D. 2009. "Flexible Global Software Development (GSD): Antecedents of Success in Requirements Analysis," *Journal of Global Information Management,* (17:1), pp. 1-31.

Yu, X., and Petter, S. 2014. "Understanding Agile Software Development Practices using Shared Mental Models Theory," *Information and Software Technology* (56:8), pp 911-921.

Zimmer, J.A., 2003. "Graph theoretical indicators and refactoring," in *Extreme Programming and Agile Methods-XP/Agile Universe 2003*, Springer Berlin Heidelberg, pp. 62-72.

APPENDIX B

Appendix B1

Key Terms:

- **Acceptance-routinization gap** can be defined as the difference between the patterns of cumulative assimilation events of an innovation across adopters at the acceptance stage and those at the routinization stage.
- **Agile acceptance** is the introductory adoption and employment of agile methods as a process innovation for one or more software development projects (Vijayasarathy and Turk 2012; Wang et al. 2012). We theorize about the use of ASD method, assuming the full use.
- **Agile methods** refer to process frameworks that are used by practitioners to develop software such as Scrum and eXtreme Programming.
- **Agile techniques** are sub-practices of agile methods
- **Agile routinization** is the usage of agile methods as a normal activity in the organization; the innovation is no longer considered out of the ordinary (Wang et al. 2012).
- **Agile software development (ASD)** can be defined as a software development team's ability to respond to changing user requirements through a process of continual readiness (Conboy 2009; Lee and Xia 2010).
- **Agile software development (ASD) assimilation** is the extent to which the use of ASD methods diffuses across the organizational projects or work processes and becomes routinized in the activities of those projects and processes (Cooper and Zmud 1990; Wang et al. 2012).
- **Assimilation gaps** can be defined as the difference between the patterns of cumulative assimilation events of an innovation across a population of adopters (Fichman and Kemerer 1999).
- **Competing Values Model**
    a. Group culture (change and internal focus) is primarily concerned with human relations and flexibility. Belonging, trust, and participation are its core values. Effectiveness criteria include the development of human potential and member commitment.
    b. Developmental culture (change and external focus) is future-oriented, considering what might be. The effectiveness criteria emphasize growth, resource acquisition, creativity and adaptation to the external environment.
    c. Hierarchical culture (stability and internal focus) is oriented toward security, order, and routinization. It emphasizes control, stability and efficiency through the following of regulations.

d. Rational culture (stability and external focus) is achievement-oriented, focusing on productivity, efficiency, and goal achievement.

- **Diffusion of innovation theory** (DIO) refers to "the process by which an innovation is communicated through certain channels over time among the members of a social system" (Rogers 1962).
- **Innovation assimilation (IA)** is defined as the extent to which the use of an innovation diffuses across "the organizational projects or work processes and becomes routinized in the activities of those projects and processes" (Purvis, Sambamurthy, and Zmud 2001).
- **Organizational culture** is "a pattern of shared basic assumptions learned by an [organization] as it solves its problems of external adaptation and internal integration, which has worked well enough to be considered valid and, therefore, to new members as the correct way to perceive, think, and feel in relation to those problems" (Schein 1985).
- **Organizational structure** is "the logically consistent clustering of an organization's elements as it searches for harmony in its internal processes and consonance with its environment" (Mintzberg 1980).
- **Perceived ASD success** is the extent to which an organization's ASD project(s) meets technical goals, remains within the budget, is delivered in time, and is accepted by the end user (Jiang, Klein, and Pick 2003; Procaccino and Verner 2006).
- **Process innovation is** a process that is perceived as new by individuals or other units of adoption (Rogers 1983).
- **Six-stages of innovation assimilation:**
    Adoptive stages
    7. *Initiation*: need for change is recognized, a match is identified between an innovation and its application in the organization
    8. *Adoption*: a decision is made to adopt an innovation
    9. *Adaptation*: an adaptation to suit the contextual needs

    Post-adoptive stages
    10. *Acceptance*: use of the innovation
    11. *Routinization*:c an increase in the extent and intensity of use (i.e. usage of the innovation is encouraged as a normal activity)
    12. *Infusion*: increased usage in a more comprehensive and integrated manner results in increased effectiveness of systems development (i.e. the innovation penetrating deeply into an organization).
- **Strong matrix** – an organizations have many of the characteristics of the projectized organization, and have a full-time project managers with considerable authority and full-time project administrative staff.
- **Weak matrix** – an organizations maintain many of the characteristics of a functional organization, and the role of the project manager is more of a coordinator or expediter.

APPENDIX B2 – SAMPLE INTERVIEW QUESTIONS

Table B2.1. Sample of Interview Questions:

History

1. Why did you choose agile?
2. What events led to this choice?

Implementation

1. Tell me about the last project you did using agile.
2. In your opinion, how did the project go?
3. What were some of the strengths and weaknesses of using agile?

Outcomes

1. Was the project considered successful? Why?
2. How did the project finish in terms of time, scope, and budget?

Individual Mindset

1. How would you define agile?
2. What was your mindset toward agile methodologies before this project?
3. What is your mindset toward agile methodologies now?

Organizational Mindset

1. What is the company's philosophy concerning the method of software delivery?
2. How is the mindset for agile (throughout the organization) moving forward?
3. Did management buy into agile? Why or why not?
4. On a scale of 1-10, how would you describe …
   a. Senior management's commitment
   b. Functional organization's commitment
   c. IT Management's commitment
   d. Developer's commitment
   e. Project management's commitment
   f. Quality Assurance's commitment
   g. …
5. What is the mindset for the clients? Vendors?
6. What do you think would hinder the sustained use of agile methods moving forward?

Company culture and structure

1. Describe the company culture.
2. How are the reporting structures arranged?
3. How would you describe the company's acceptance of change (e.g. quick to embrace change, resistant to change, etc.)?
4. How would you describe the management style of your direct manager(s)?

APPENDIX B3

History

Prior IS literature refers to history as "the continued influence of past choices" (Hirschheim and Klein 2012). As a process innovation, many agile methodologies are introduced into organizational settings with a history of social norms and enduring ISD processes. A given organization's history may include a bevy of prior projects that undoubtedly affect its subsequent choices. Thus, it is imperative to understand the antecedent conditions that inform these choices (Newman and Robey 1992).

Context

Although corporate context has not been defined using a fixed a set of universally agreed upon elements, corporate context has been studied as the circumstances in which organizational processes take shape (Burgelman 1982) In ISD, the corporate context envelopes the implementation processes described by the antecedents and the resulting outcomes (Newman and Robey 1992; Stein and Zwass 1995). In our model, factors such as its corporate culture and structure, the industry in which a company competes, and its business strategy are used to describe each organization's corporate context. Prior research shows that organizational context can be differentiated along a multitude of factors (Larson and Gray 2010), with one of those being cultural values (Denison and Spreitzer; Cameron and Quinn 2011). Thus, we focus on the relationship between different corporate culture orientations and the adoption of agile methods. Drawing upon

prior ISD research, we utilize the competing values model to differentiate opposing corporate values (Iivari and Iivari 2011).

ASD Acceptance (Initial Use)

ASD acceptance refers to the initial employment of ASD methods as a process innovation (Vijayasarathy and Turk 2012; Wang et al. 2012). The introduction of agile methodologies involves the appropriation of a particular agile method, which guides work activities. Within an ASD context, a group of cross-functional stakeholders may work together to produce an IT artifact. Therefore, we focus on the socio-technical factors involved during this process including critical incidents, conflicting interpretations, and critical factors of success and failure. Our model examines the effects of the ISD process on project outcomes.

Project Outcomes

Project outcomes refer to the perceived success or failure of an agile implementation. Drawing upon the social process model, we studied agile implementation success outcomes as the result of a series of events over time as perceived by interviewees rather than the amount of variation explained in a dependent variable (Newman and Robey 1992). Therefore, we interpreted the success of agile implementations based on both stakeholders' perceptions of the quality of given agile implementation as well as how well agile outputs met predefined specifications of time, cost, and scope constraints. In contrast to factor models, process models provide a story that details the association between antecedents and outcomes. Thus, we examined the social processes that

impacted the antecedents and ISD process in each agile implementation, recording project outputs such as lessons learned, new histories, and working software.

Routinization Decision

ASD routinization refers to usage of agile as a normal activity. At the routinization stage, an increase in the extent and intensity of use of an innovation is encouraged as a normal activity (Senapathi and Srinivasan 2012). We examine the intentions of normal use after each agile implementation while closely examining any gaps between the acceptance and routinization stages.

APPENDIX B4 – SAMPLE DATA ANALYSIS

Table B4.1. Beta's Conflict with Agile
Organizational arrangement
Working in cross-functional agile teams, for some, meant become more of generalist than a specialist. For example, developers were divided by their coding areas of front end and back end, and even by the technology they specialized in such as .NET. Beta's agile asked developers to work on both the front and back end of the code as well as share specialized knowledge with team members.

A lot of people in the company have put them into them niche according to their technical ability and also their functional one. It would be asking a lot of people to get out of their comfort zone.

-Beta.Dev8

Contribution: Developer despecialization

Thus, the introduction of agile methodologies in many ways was a disruption of Beta's normal ISD working patterns. This caused some developers outside of Beta's agile team to push back as they feared that the expansion of agile would weaken the contribution of their specialized skillset.

We are really asking to change what they have done. The people who have just come in here within the last 4-5 years that haven't really been completely indoctrinated in to Beta, the way that is. It is a stretch whenever people take these different kinds of jobs. I come from software development and you walk into a place like this and you say well ok, **I'm going to have to be able to be ok with my skillset going away**. Unless you have some other reason for doing that it is kinda difficult to have someone do that. The people that have been here forever to ask them to go back and learn all this stuff, it seems like a lot. - Beta.Dev1

Lack of openness to altering working structure – Departmental to dedicated teams There are few people in the organization who are opening their minds to concepts of dedicated co-located teams focused on work, which is different than what has been our standard model for executing projects. It feels like decades, the culture is so ingrained here, of being more focused on resource utilization than throughput. it feels like our culture here is, we would rather have a hundred different developers working on 10 different projects at a time then having all of our developers working on one project and getting it done. –Beta.Dev4
Departmental resistance to projectized work structure IT, in my opinion, are the biggest resistors right now because they are the ones that are going to stand the most change. There are going to be significant impacts if we see this transformation through to resource and reporting structures job titles that kind of thing –Beta.Dev4
Pushback encounter One of my fears, I am actually running into this right now, a couple of resource managers want take some of the resources back. That will completely impact our velocity, change up the team. It will be a bump we will have to get over. I hear rumors that they would like to put in model where you have partially allocated resources, not doing co-location. That would make this much more difficult and we are going to slip back into waterfall real quick if we start doing those patterns. -Beta.PM

Adoption to adaption gap – Lack of facilitating project culture within hierarchical cultures

Following diffusion innovation's theory, we follow the diffusion of agile methodologies through the assimilators that house ISD practices within their existing culture and structure. In this study, each of the four companies possessed a dominant philosophy for ISD. Of these, Alpha, Beta, and Century encountered issues with their existing waterfall methodologies, which led to their consideration of agile alternatives and their eventual decision to try agile methods. However, Alpha failed to alter its existing organizational structure, which conflicted with agile workflows. Thus, Alpha's actions can be identified and examined as an adoption-to-adaption gap. Alpha's low quality implementation led to multiple deviations from best practices.

In comparison to Beta, Alpha's poor execution seem to be due, at least in part, to the organization's inability to facilitate a supporting culture. Alpha.ISDMgr3 commented,

> Towards the end... I thought it was great and we did it wrong, but it never changed my mind that it would be a good match in other circumstances. I heard other PMs were left with a really bad taste in their mouth.

Thus, in line with prior research, we suggest that hierarchical organization's that choose not to alter their overall organizational structure may create a new, albeit temporary, team structure to insulate the project and facilitate agile methods. In this study, Beta was able to create this kind of structure by dedicating and insulating its project teams. In addition to providing a facilitating culture for the methodology, methodology discipline should be

ensured by training and oversight by outside coaching. Compared to Alpha's implementation quality, Beta's high degree of methodological discipline reduced the number of deviations from best practices and led to a high degree of perceived project success.

Routinization to infusion gap - Lack of formalization

Both Century and Dynamic have enjoyed a history of successful ASD implementations. The companies share many similarities such as their mostly developmental cultures, relatively low average age for employees, and external ISD practices. One observable difference in their assimilation of agile methods is the rate in which they adopted agile best practices. For Dynamic, agile has been the only methodology that company has used for ISD, whereas Century used waterfall methods for many years before trying agile. Interestingly, after more than five years of implementing a customized flavor of XP, Dynamic decided to adopt more standard agile techniques, which includes best practices that company decided not to implement previously such as minimizing task switching. Despite the company's history of successful agile implementations, none of its stakeholders received formal training in agile methodologies, though most of its employees have read book chapters on agile methodologies. In comparison, Century sent to most of its ISD stakeholders to agile training and put in place more formalize processes of understanding the Scrum methodology. Thus, we observed a faster diffusion rate from routinization to infusion between to Century and Dynamic, 2 years to 5 years respectively. In comparison, Century's higher degree of formalization fueled a higher degree of methodology knowledge. In fact, some of Century's key informants indicated

that agile training was a key differentiator between individuals in ISD that had fully adopted agile, and those that had yet to fully adopt an agile mindset. When asked what would be the biggest hindrance to the continued use of agile at the company, Century.Dev2 answered,

> Getting everyone up to speed or trained on the processes. For five of us, that was probably our first project. We have a few other project that we are doing agile on. I think it is getting everyone used to how to do this the right way. (2/3) The other 1/3 would be getting projects that we could actually do this on (in terms of size and customer buy in).

Concerning the few developer's that had yet to buy into agile methodologies, Century.Dev1 pointed to their lack of training with,

> We have some developers that have just been developing for a very long time and are not necessarily (onboard) -- I think a large part of it is the training. They have not had the full amount of training yet to understand why we are doing it and its necessity and need for us.

In the end, since both Century and Dynamic employ democratic methods to recruit more stakeholder buy-in. Thus, the adoption of an innovation may be less mandatory in these companies than companies that contain a more top-down approach. Although Dynamic encouraged the experimentation with optimal best practices, the company was slower to mandate specific processes as "the way" to do things. In contrast, with agile, Century took a more formalized approach after the success of its first agile project, sending most of its stakeholders to formal training. As a result, Century saw a faster diffusion of agile methodologies than Dynamic who approaches diffusion by introducing agile values into the culture, but not training on any particular method.

VITA

Corey Wayne Baham, a native of Baton Rouge, LA, received his bachelor's degree at the Louisiana State University in 2004. Thereafter, he worked in various IT roles, and taught Computer Science at the secondary level. He later earned his master's degree in August 2011, and is a candidate to receive his Ph.D. in May 2016. In August 2016, he will began as an Assistant Professor of Information Systems at Oklahoma State University.