HAVE YOUR CAKE AND EAT IT TOO? SIMULTANEOUSLY PURSUING THE KNOWLEDGE SHARING BENEFITS OF AGILE AND TRADITIONAL DEVELOPMENT APPROACHES

Alec W. Cram and Marco Marabelli

Abstract

This research examines how organizations manage knowledge sharing processes in systems development projects that employ both agile and traditional development techniques. Using a longitudinal case study, we draw on one company's experience with a system implementation that employed a traditional approach during its first phase and then a hybrid, agile-traditional approach in its second phase. By applying an ambidexterity theory lens, we find that the adoption of a hybrid approach allowed the project to continue to exploit the traditional techniques that were working well, abandon techniques that were underperforming, and explore the use of agile techniques in selected areas.

Keywords: Agile development, traditional development, knowledge sharing, ambidexterity, systems development

HAVE YOUR CAKE AND EAT IT TOO? SIMULTANEOUSLY PURSUING THE KNOWLEDGE SHARING BENEFITS OF AGILE AND TRADITIONAL DEVELOPMENT APPROACHES

Introduction

The increasing popularity of the agile approach to systems development has significantly altered the activity of planning, designing, and implementing software within many organizations by shifting the focus away from a structured, traditional approach to information systems development (ISD) (Dingsøyr et al., 2012, Nerur et al., 2005, Vinekar et al., 2006, Wang et al., 2012b). The attention being paid to the agile approach has contributed to growing scrutiny related to the continued use of traditional development approaches (e.g. waterfall), which advocate a more formal, linear development style with a focus on documentation and pre-defined stages (West and Grant, 2010, Royce, 1970, Boehm, 1976).

However, it remains unclear if ISD projects perform better using a 'pure' systems development approach that closely adheres to one set of principles (e.g., agile *or* traditional) versus the integration of a collection of diverse techniques together in a hybrid approach (e.g., agile *and* traditional). Although organizations routinely adapt their ISD approaches from a pure, 'by-the-book' interpretation (Fitzgerald et al., 2006, Mahadevan et al., 2015, Wang et al., 2012a, Cao et al., 2009a, Port and Bui, 2009, Cram and Newell, 2016), projects that employ a mix of agile and traditional techniques are sometimes perceived as being inconsistent in their communication, control, and role assignment (Nerur et al., 2005, Boehm and Turner, 2005). Despite this, such hybrid approaches are becoming common for companies with a history of traditional development, but that also have an interest in adopting a more modern, flexible approach (West and Grant, 2010, Cram and Newell, 2016). Although hybrid approaches can be successful (Port and Bui, 2009), it is a challenge to reach a point where a fusion of agile and traditional techniques can work together effectively (Boehm and Turner, 2005, Nerur et al., 2005, Vinekar et al., 2006, Mahadevan et al., 2015). This is particularly difficult when project teams are trained solely in conducting traditional software development, but intend to introduce agile components incrementally into their ongoing program of ISD.

Our research aims to address this important and practical challenge by examining *how* organizations with limited agile experience navigate the transition from a traditional ISD approach to a hybrid, agile-traditional approach. In particular, we focus on the implications for knowledge sharing because of its important role in driving project success (Chau et al., 2003, Melnik and Maurer, 2004, Chan and Thong, 2009) and because, as the literature clearly indicates, knowledge sharing processes are conducted very differently on the basis of the ISD approach chosen (Cabral et al., 2009, Nerur et al., 2005, Crawford et al., 2006, Ghobadi, 2015, Conboy et al., 2011, Ghobadi and Mathiassen, 2015, Alzoubi et al., 2016). In particular, we focus on knowledge sharing-related factors that need to be considered by an organization in order to effectively shift from a pure agile approach to a hybrid approach, as these factors have not yet been identified in the literature. Therefore we pose the research question: *How do knowledge sharing processes associated with ISD projects change when agile techniques are increasingly used alongside traditional techniques?*

We address our question through the use of longitudinal, qualitative fieldwork at a company where staff had deep experience in traditional approaches, but minimal practical experience with agile. Following a first, unsuccessful implementation of a Customer Relationship Management (CRM) system, management initiated a second project phase (referred to as the 'relaunch'), which incorporated selected agile techniques and resulted in a fusion of the traditional and agile approaches (i.e., hybrid). Therefore, it was meaningful to observe the challenges associated with the shift from a traditional to a hybrid approach in a context where the development team needed to learn *how* to manage knowledge sharing processes during the relaunch. Through a rigorous analysis of our collected data, we were able to identify a series of *factors*, related to organizational mechanisms; *patterns*, associated with the evolution of the adoption of knowledge sharing processes while transitioning from agile to hybrid; and organizational (internal) and environmental (external) *themes* that are relevant and useful to staff not specifically trained in agile approaches, but who are undertaking an ongoing, incremental shift from a traditional to a hybrid ISD approach.

While our paper provides meaningful relevant practical implications for managers, we also contribute to theory. Building on previous research (e.g., Cao et al. 2009a; Ramesh et al. 2012; Vinekar et al. 2006), we suggest that the ability to manage the conflicting demand of efficiency (by adopting established best practices related traditional approaches) and flexibility (by being able to manage fast-changing implementation issues typical of agile approaches) can be examined using an ambidexterity lens; our findings suggest that emerging ambidextrous capabilities, here defined as those abilities needed to face the ongoing (and often unpredictable) demand to explore and exploit knowledge, are essential for 'learning by doing' in hybrid implementations. In addition, our identification of a series of factors, patterns, and themes related to the increasing use of agile techniques alongside traditional techniques and their impact on knowledge sharing processes contributes to developing the important links that exist (but that have not yet been formally defined) between the ambidexterity and ISD literature.

Our paper is structured as follows: In the next section, we discuss the research background and theory base. Next, we outline our methodology, including details of the data collection and analysis. The results are then presented and our findings are discussed using an ambidexterity lens. We conclude by presenting a series of future research opportunities.

Background and Theoretical Base

The Transition from Traditional to Agile ISD

Consistent with past research (Iivari et al., 2000, Huisman and Iivari, 2006, Cram and Brohman, 2013), we distinguish between the concept of a development approach (i.e., the high level goals and principles of systems development, such as agile or traditional), development methodology (i.e., a grouping of guiding development concepts, such as eXtreme Programming or Scrum), and development technique (i.e., a lower level activity conducted as part of a development project, such as pair programming). Traditional software development approaches are oriented around a pre-defined, incremental sequence of steps beginning with the analysis of system requirements, followed by the design, development, implementation and maintenance of the system (Royce, 1970, Boehm, 1976). The benefits of the approach include its straightforward, linear design and clear milestones that are helpful to manage and monitor the progress of a project (Hughey, 2009). Traditional approaches commonly rely on formal techniques including detailed procedures, output, and approvals, which are shared amongst team members through a variety of documents related to the project (Alavi and Leidner, 2001a).

In comparison, an agile approach draws from a set of principles set forth in the Agile Manifesto (Beck et al., 2001), including advocating for face-to-face team interactions, collaboration, flexibility to respond to changes, and attention to excellence. Typical development techniques adopted by agile teams include pair programming (i.e., two developers working together on the development and refinement of a piece of code), stand-up meetings (i.e., short, daily meetings with project team members), story cards (i.e., short descriptions of desired system functionality), planning poker (i.e., a team exercise to arrive at consensus for the amount of effort required for a task), and sprints (i.e., iterative cycles of work typically lasting from 2-4 weeks) (Balijepally et al., 2009, Dingsøyr et al., 2012).

Although the proportion of organizations adopting a primarily agile approach has risen significantly in recent years, figures also suggest that agile techniques are regularly blended into existing traditional techniques to form a hybrid approach (West and Grant, 2010, Serrador and Pinto, 2015). This allows for the customization of low-level development techniques to the needs of the company, while also smoothing the transition away from an institutionalized, traditional development approach. Although past studies have widely examined hybrid approaches (Boehm and Turner, 2005, Nerur et al., 2005, Vinekar et al., 2006, Mahadevan et al., 2015) this research provides somewhat 'static' findings, which are helpful in understanding why hybrid works in practice, yet unveiling few details into *how* organizations navigate the transition from a traditional approach to a point where they are able to use (and hopefully benefit from) a mix of traditional and agile approaches, especially with respect to knowledge sharing processes. In particular, it is relevant to understand how organizations with staff not specifically trained in agile are able to transition to such an approach using 'learning by doing' techniques; namely, led by an agile savvy project manager, the team faces the challenge of adopting a collection of techniques that are typical of an agile approach. The extremely scant literature in this regard begs the question of *how* organizations transition away from traditional development to adopt a hybrid approach. We attempt to explore this issue by focusing on the dynamic unfolding of practices leading to a shift from the 'status quo' (a pre-determined, traditional approach) to a situation where a hybrid, agile-traditional approach is undertaken. To this end, we aim to uncover insights related to ongoing knowledge sharing processes that are associated with such a combination of ISD approaches.

We suggest that it is important to examine this dynamic, focused perspective because organizations routinely experience practical challenges during a transition to hybrid. Particularly when past ISD projects have been rooted in their adherence to techniques such as written documentation (agile minimizes it), unchanging product requirements (agile encourages ongoing changes), and minimal direct contact with customers (agile encourages extensive face-to-face interactions), managers are faced with difficult decisions on what development techniques should remain traditional and what techniques should transition to agile. Companies adopting a hybrid approach are challenged to find a way to balance these seemingly conflicting areas without sacrificing the benefits associated with a 'pure' methodology (i.e., where all techniques complement one another); however, by not adhering strictly to either traditional or agile, organizations attempt to gain benefits of both approaches, but may unintentionally make trade-offs that erode the core principles of both approaches to the point that neither performs effectively. We address this practical challenge, which also reflects a theoretical gap, by examining one key aspect of software development, knowledge sharing, which we discuss next.

Knowledge Sharing in Systems Development Projects: Principles and Ambidextrous Capabilities

Knowledge sharing is viewed as an important set of processes that can contribute to effective systems development projects. It is widely recognized that knowledge sharing processes vary substantially under a traditional versus agile approach (Nerur et al., 2005, Ghobadi, 2015, Ozer and Vogel, 2015, Boehm and Turner, 2003, Cao et al., 2009a). In an ISD context, knowledge sharing refers to the transfer of both tacit and explicit knowledge amongst project stakeholders. This includes project documentation, user requirements, training, developer interactions, and management guidance. Past research distinguishes between knowledge viewed as an 'object' and can be exchanged in written form (i.e., consistent with traditional development approaches), as compared to knowledge that is seen as a 'relationship' that is exchanged between project members through daily interactions (i.e., consistent with an agile development approach) (Alavi and Leidner, 2001a, Newell et al., 2009, Nerur et al., 2005, Feldman and Orlikowski, 2011, Schatzki et al., 2001, Sandberg and Tsoukas, 2011). However, the ISD literature points to the challenges associated with both perspectives (Boehm and Turner, 2003, Vinekar et al., 2006); on one hand, using written documentation represents a straight-forward approach, where well trained developers operate in an efficient manner by relying on consolidated best practices. However, this makes it difficult to manage unforeseen issues and generally does not allow for ongoing changes and departures from planned development strategies, therefore creating rigidity. On the other hand, focusing on informal communication, face-to-face meetings, and knowledge sharing through social practices can create a more flexible and unstructured environment – yet it requires the awareness and ability to adapt to emerging circumstances (Boehm and Turner, 2004). In fact, while valuable insights have been identified in both approaches to knowledge sharing, the literature has not delved yet into issues related to the factors that managers should consider when deciding how to most effectively share knowledge on projects where both traditional and agile techniques are used (Ghobadi and Mathiassen, 2015, Joshi et al., 2007, Ozer and Vogel, 2015, Kotlarsky and Oshri, 2005, Oshri et al., 2008). Recent literature has not yet focused on these knowledge-centric 'transitions' involving a shift from traditional to agile approaches in *ongoing* ISD processes. Addressing this theoretical gap is extremely relevant for practitioners, particularly those with minimal agile experience, due to the importance of effective knowledge sharing in achieving project success, as well as the increasing adoption of hybrid approaches. Refer to Appendix D for further details on the key goals and contributions from past research, including the specific elements that are unique about our research.

Comparing Knowledge Sharing in Agile Versus Traditional: An Ambidextrous Approach

By building on (and extending) prior ISD literature that acknowledges the conflicting demand of efficiency (traditional approach) and flexibility (agile approach) (Cao et al. 2009b; Vinekar et al. 2006), we attempt to address this gap using an ambidexterity lens, which considers the ability to manage the conflicting demand of exploring new products, services and competences, while exploiting existing ones (Katila and Ahuja, 2002; He and Wong, 2004). Specifically, March (1991) argues that "exploration includes things captured by terms such as search, variation, risk taking, experimentation, play, flexibility, discovery, innovation. Exploitation includes such things as refinement, choice, production, efficiency, selection, implementation, execution" (p. 71). Within the ambidexterity literature, some scholars point to the central role that knowledge sharing plays in pursuing exploratory, as well as exploitative activities (Durcikova et al., 2011, March, 1991, Newell, 2015). To this end, ambidexterity capabilities are associated with a firm's ability to balance between exploiting the use of existing knowledge, while also exploring new knowledge in order to remedy deficiencies (Turner et al., 2013). In order to achieve long-term success, ambidexterity enables continual adaptation over time in response to both small and large changes in strategy, culture, and structure (Tushman and O'Reilly, 1996, Raisch et al., 2009).

Indeed, the suggestion that different knowledge sharing capabilities are required depending on the adoption of either a traditional or agile development approach is consistent with the fundamental tension between knowledge exploration and exploitation (Cao et al., 2009b, March, 1991). In an ISD context, exploitation refers to the use of existing knowledge, such as at a firm with expertise pertaining to a traditional development approach, while exploration is increasingly focused on exploring new knowledge, such as that created through iterative, collaborative interactions between agile team members. Past literature finds that utilizing existing knowledge leads to efficiency, while exploring new knowledge leads to flexibility (Adler et al. 1999). However, it is worth noting that each organization's development approach is unique, whereby both traditional and agile approaches are likely to incorporate at least some degree of both knowledge exploration and knowledge exploitation. Nevertheless, by building on the ISD literature that points to the exploration/exploitation dilemma, while contrasting traditional and agile approaches (e.g., Cao et al. 2003; Cao et al. 2009b; Ramesh et al. 2012; Vinekar et al. 2006), we suggest that in the former approach, the tendency to address efforts to knowledge exploitation processes is prevalent, while in the latter approach, the focus shifts towards knowledge exploration processes. Therefore, an effective hybrid approach will require ambidextrous capabilities.

Over the past decade, the popularity of agile development has given rise to the application of ambidexterity principles, as companies seek to explore the speed and flexibility of agile methods, while continuing to exploit traditional methods (Ramesh et al., 2012, Vinekar et al., 2006). The ISD literature mainly focuses on companies that pursue ambidexterity by developing independent capabilities in both approaches, but then select only one 'pure' approach for use on a development project (Vinekar et al., 2006).

However, it is important to note that some firms, as in our case study, simultaneously integrate development techniques of both approaches together within the same ISD project. Such firms adopt a hybrid development approach that concurrently employs traditional techniques (e.g., extensive planning and documentation), as well as agile techniques (e.g., pair programming, story cards). This offers elements of an approach that is proven and reliable (i.e., traditional), while simultaneously taking advantage of an innovative alternative (i.e., agile). Even with the potential for the conflicting demands inherent with hybrid approaches, including clashes with an organization's culture and social norms (Walsh and Seward 1990), past studies show that they have the potential to adapt, make trade-offs, and result in successful outcomes (Fitzgerald et al., 2006, Cao et al., 2009a, Ramesh et al., 2012). However, despite the links between ambidexterity and performance (Ramesh et al., 2012), relatively little is known about how managers decide to choose one ISD technique over another. For example, consider a manager at a company that has a lengthy history with traditional development and little agile experience, but has a growing interest in introducing an agile approach. The manager is responsible for selecting the collection of techniques that will be used on an upcoming project and can decide to continue to use a) mostly traditional techniques alongside a few agile techniques, b) a balanced mix of traditional and agile techniques, or c) a few traditional techniques alongside a heavy dose of agile techniques. Although the manager is motivated to choose the best alternative to enhance project performance, the team's lack of experience with agile makes the choice difficult. When approaching this practical challenge from an ambidexterity perspective, the manager would be expected to exploit the traditional techniques that could provide continued value on the project, while exploring the potential of agile techniques to make a unique and valuable contribution to the project. However, the current literature has not yet considered what specific factors the manager should consider when making this determination in order to achieve a successful transition.

In this section we provided an overview of knowledge sharing in traditional versus agile systems development, including an introduction to how ambidexterity principles can help to bridge the differences between the approaches. We suggest that a gap exists in the literature regarding the lack of understanding of what factors managers should consider when choosing between agile and traditional techniques during a transition from traditional to hybrid and when staff are not ‘agile savvy’. Due to the importance of knowledge sharing and the growing pervasiveness of hybrid development approaches, we aim to clarify how knowledge sharing processes associated with ISD projects change when agile techniques are increasingly used alongside traditional techniques. We use this line of inquiry as a guide to the data collection and analysis of our case study, which we discuss next.

Methodology

In order to empirically study knowledge exploration and exploitation in hybrid approaches, we draw on Chau et al. (2003), who propose a framework that evaluates the underlying characteristics of knowledge sharing processes in both traditional and agile systems development. The framework identifies and compares the characteristics of eight key knowledge sharing processes (see Table 1 below). Despite its publication when agile was relatively new, the framework remains useful as a template to identify the fundamental differences between knowledge sharing in a traditional and agile approach, as evidenced by regular citations in more recent systems development literature (Crawford et al., 2006, Henderson-Sellers and Serour, 2005, Qumer and Henderson-Sellers, 2008, Ghobadi and Mathiassen, 2015). However, in order to further update the framework to the most current point of view, we revisited the ISD literature to confirm if the processes remained relevant and to add additional details to supplement their underlying characteristics. We found support for the eight knowledge sharing processes proposed by Chau et al., but updated the descriptions of the characteristics associated with a traditional and agile approach. We also list a series of updated references that support these changes in Table 1.

Knowledge Sharing Process	**Traditional ISD Approach**	**Agile ISD Approach**	**Supporting References**
Documentation	Extensive documentation, consisting of requirements, design specifications, development plans, etc.	‘Just enough’ documentation, which may include techniques such as story cards.	Balijepally et al. (2006), Nerur et al. (2005)
Requirements and Domain Knowledge	Formalized requirements captured before initiation of design and development; as-needed interaction between development team and customers.	Active stakeholders and extensive user participation throughout the project, including a high degree of readiness for change. Requirements are estimated for workload, prioritized, and contextualized as stories or test cases.	Boehm and Turner (2003), Cao et al. (2009a)
Training	Formal, facilitated training sessions. Often conducted in classrooms using static training materials.	Informal training practices to enhance knowledge sharing, such as pair programming and daily stand-up meetings.	Cao et al. (2009a), Fruhling and De Vreede (2006)

Competence Management	Formal status reports, assigned responsibilities based on document ownership, direct managerial oversight.	Ongoing communication between stakeholders to establish a shared understanding and to discuss progress. Collective code ownership allows team members to monitor the quality of their colleagues' code.	Batra (2009), Cao et al. (2009a), Maruping et al. (2009)
Trust and Care	Low reliance on trust, due to establishment of formal policies, including a stage gate process that mandates periodic management review.	High reliance on empowerment and trust within the team, built from techniques such as collective code ownership, stand-up meetings, collaborative workspaces and pair programming.	Boehm and Turner (2003), Cao et al. (2009a), de Cesare et al. (2010)
Team Composition	Clearly defined, role-based teams, such as business analysts, developers, and testers.	Cross-functional teams, with team members playing multiple roles throughout the project. Standard 40 hour workweeks are employed to preserve work-life balance.	Balijepally et al. (2006), Dyba and Dingsoyr (2008), Maruping et al. (2009)
Continuous Learning	Post-mortem reviews at the end of project stages or at project completion.	Person-to-person interactions during development, using techniques such as pair programming and feedback sessions. Retrospective activities at the end of sprints to review success factors, lessons learned, and obstacles.	Fruhling and De Vreede (2006), Fitzgerald et al. (2006), Karlsson and Agerfalk (2009)
Knowledge Repositories	Heavy reliance on explicit knowledge stored in documents within formal knowledge repositories.	Reliance on tacit knowledge, trial and error, and communication among team members. Use of lightweight, informal knowledge repositories in either non-digital (e.g., storyboards) or digital (e.g., LeanKit) form.	Balijepally et al. (2006), Boehm and Turner (2003), Fitzgerald et al. (2006), Maruping et al. (2009), Nerur et al. (2005), Vinekar et al. (2006)

Table 1. Agile-Traditional Comparison of Knowledge Sharing Processes (adapted from Chau et al., 2003)

Data Collection

This research employs a qualitative, longitudinal case study approach at a technology recruiting firm we refer to using the pseudonym TechRecruit. Over a 21-month period, we collected data via 32 interviews with company executives, project team members, and end users, as well as 48 company documents. Single site case studies are commonly used within agile research, such as in Fitzgerald et al. (2006), Moe et al. (2010), and Persson et al. (2012), as well as more generally in IS research, including recent examples such as Huang et al. (2017), Aaltonen and Tempini (2014), and Guillemette et al. (2017). Past commentators, such as Benbasat et al. (1987) and Yin (2009) argue that single case studies can provide valuable insights for exploratory studies, such as this research. In addition, the IS literature is particularly supportive of single case studies applied to organizational challenges associated with learning processes that are needed to ‘understand’ new technologies and processes. These aspects are directly related to our research objective: examining the emergence of knowledge sharing processes arising from *ongoing* learning of developing systems with respect to hybrid development approaches. For example, single cases were applied to learning processes related to the emergence of the Enterprise Resource Planning software, including (Bose et al., 2008, Elbanna, 2007, Newell et al., 2003, Scott and Wagner, 2003).

Our study examines the design and implementation of a CRM system that was rolled out in two project phases; the first phase employed exclusively traditional development techniques and the second phase employed both agile and traditional techniques together (more details on the organization and project are presented in the next section). Our data was collected shortly after the conclusion of each phase: 22 interviews were conducted during March and April 2013 and an additional 10 interviews during May-November 2014 (please refer to a timeline in Figure 1). Although we recognize that it may have been preferable to conduct the interviews during the completion of the phases themselves, TechRecruit management requested our participation shortly afterwards, due to project time pressures. Our longitudinal approach helps to capture the changes to knowledge sharing that resulted from adding agile techniques to a project that had previously relied on traditional techniques. A total of 28 TechRecruit employees participated in the 32 interviews, as some individuals were interviewed on multiple occasions and two interviews comprised multiple participants. The duration of interviews ranged between 30 and 70 minutes in length. Interviews were conducted using a semi-structured approach, which focused around the activities conducted during each project phase, including the knowledge sharing processes and systems development techniques in place. Additional questions were posed to the interviewees to gather examples of events that positively or negatively affected their view of the project and resulting system, in order to better understand the knowledge sharing opportunities and challenges.

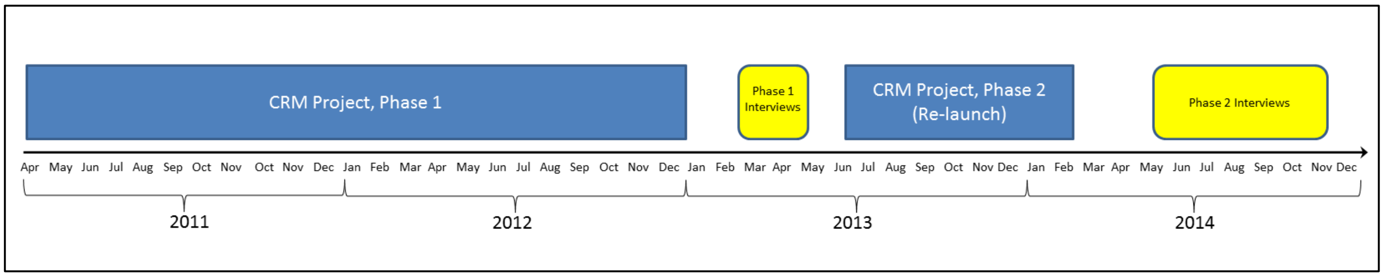

Figure 1. Project and Data Collection Timeline

Participants were selected based on their participation in the CRM project either as company executives (e.g., CEO, CIO), business management (e.g., Accounting Director, Marketing Manager), IT management (e.g., IT

Project Manager, IT Director), or end users (e.g., Financial Analyst, Sales Support Specialist). After the meeting with top management, we followed a snowball sampling method (Rankin and Bhopal, 2001) by identifying new people to interview based on the recommendations of existing participants. We attempted to supplement our TechRecruit interviews by also interviewing vendor personnel. Despite receiving TechRecruit's approval to do so and discussing the project with vendor management, the vendor ultimately decided not to participate in this study.

A series of company documents were also collected to aid in data triangulation, which refers to the repeatability of an observation in order to arrive at more accurate findings (Yin, 2009, Stake, 2006). A total of 48 documents were collected that related directly to the project and consisted of emails, reports, project plans, requirements documentation, sprint documentation, and training materials. We used these documents to corroborate the events discussed in the interviews and to better understand the knowledge sharing processes taking place on the project.

Data Analysis

The data analysis was undertaken by first compiling a broad narrative of the whole case, which aided in defining the timeline of events that unfolded during the two project phases (Klein and Myers, 1999). The narrative was also used by the authors to ensure a shared understanding of the case. Missing elements from the initial writing of the narrative were clarified by individuals at TechRecruit, as good access to the company allowed us to undertake several 'back-and-forth' interactions between fieldwork and analysis, in order to improve the accuracy of the data.

Next, we examined the interview transcripts and company documents using NVivo and coded the data into the eight knowledge sharing processes identified in Table 1. For example, when an interviewee discussed the process used to collect requirements from users, we coded the passage to the 'Requirements and Domain Knowledge' process. We separated the coding between the first and second project phases in order to distinguish the temporal aspect of the project and more easily enable 'before' and 'after' comparisons (Miles and Huberman, 1994). The first author conducted a trial coding exercise using a sample of the data and the results were reviewed and discussed with the second author. The remaining data was then coded and the second author reviewed the results. Further discussion was conducted and all disagreements were satisfactorily resolved. A total of 675 text segments were coded across the eight processes (471 segments in the interviews and 204 segments in the company documents). Refer to Appendix A for a detailed listing of coding results by process and source.

We then compared the collected data from the first phase of the project (i.e., employing a traditional approach) to the data from the second phase (i.e., using a hybrid, agile-traditional approach) in order to identify similarities and differences in the knowledge sharing processes. For each knowledge sharing process, we re-reviewed the coded data to determine the extent that the underlying ISD techniques remained primarily traditional, had party transitioned to agile, or had fully transitioned to agile.

Since our study is oriented around the increasing use of agile techniques, our analysis also considered the events associated with the decision to adopt agile in some areas, but not others. To do so, we adopted Langley's (1999) process theorizing approach of temporal bracketing, which calls for the separation of data into distinct periods (the first and second project phases, in our case), followed by the identification of key events that can aid in clarifying structures over time. We re-examined the data associated with each knowledge sharing process for events (e.g. management choices, project activities) that link the systems development techniques in the first project phase to the second project phase. For example, the decision to use SharePoint as a knowledge repository during the project phase would be considered an event, as would the decision to transition to the LeanKit knowledge repository during the second phase. Refer to Appendix B for a listing of the events corresponding to each knowledge sharing process in the first and second project phases. During our analysis of these events, we also reviewed the data for evidence of the root causes linking an event in the first phase to an event in the second phase (e.g. dissatisfaction with SharePoint functionality) in order to better understand the reasons behind the choice to introduce new agile techniques or to continue using traditional techniques. We present and analyze the results of this analysis in the following section, but first provide a detailed overview of TechRecruit.

Company Overview

TechRecruit is a staffing and consulting company serving the East Coast, Great Lakes area and Southeast Region of the United States. Founded in 1989, they specialize in employment recruiting for technology-related positions such as systems administrators, systems developers, business analysts, and IT project managers.

Staffing companies operate as a third party intermediary between employers and employees. The ability of a staffing company to gain and maintain competitive advantage rests on its ability to quickly find matches between the employer's needs and employee's technical and relational skills (Ward, 2004). To this end, it is crucial that staffing companies are equipped with information systems that facilitate fast and accurate data analysis and reporting capabilities. A manager at TechRecruit, in one of the initial interviews, pointed out that "*we have to do this job quickly and in a professional way*".

In the fall of 2011, the company made the critical decision to abandon its legacy CRM system, which was first adopted in 1999. Despite the software serving effectively for over a decade, TechRecruit management also felt that they needed a more modern system with cloud-based capabilities, advanced security features, greater speed, real time backup capabilities, and other technical features. For instance, an IT project manager told us that "*we needed to move to a web-based system that could be flexible and allow people to work from everywhere. If there is a snow storm, you can work from home*".

A detailed requirements-gathering exercise was undertaken by a cross-functional TechRecruit team and several replacement options were identified. Presentations were held with a series of vendors in February 2012 and a system called Super-CRM (pseudonym) was chosen. Rather than opting for one of the widely known systems such as SalesForce, Super-CRM was developed by a small, local company specializing in solutions for staffing companies. TechRecruit management chose Super-CRM because of its attention to staffing company needs and that it was substantially less expensive than other systems.

A project team was created with representatives from both TechRecruit and the Super-CRM vendor. Although a 'base' Super-CRM system had already been developed by the vendor, a variety of technical customizations were required to prepare the system for implementation at TechRecruit. These included: a) data mapping, data migration, data validation; b) infrastructure setup and configuration; c) workflow customization (e.g. job opening alerts); d) reporting customization; and e) integration with other applications (e.g. email). While TechRecruit was primarily responsible for requirements gathering, testing, and training, the Super-CRM vendor oversaw the software code changes. The project adopted a 'by-the-book' traditional development approach, drawing on waterfall principles. However, the design, development and implementation of the new system took much more time and effort than expected and the application experienced poor performance after it went live. For instance, one recruiter observed that "*the [new] system is so slow when moving from one screen to the other and it is so frustrating because we work with time pressure; this is part of our business*". The performance of a system is particularly relevant for recruiters, as they work on commission and their paycheck is associated with finding good matches quickly and effectively. The same recruiter pointed out that:

> *There is a lack of customization right now. I don't have a daily calendar on the screen. I have to put data in the system; the system then doesn't put this data on my calendar that I see every day.*

So I have to do this out of the system and this is not good. You know, contacts and relationships, networking is very important and if you forget to call back a client, you lose the relationship and the job! – Recruiter

Given that almost a year was devoted to customize the system for the first phase of the project, both management and staff were dissatisfied with the project outcome. Despite some minor improvements that were made during the first quarter of 2013, a second project phase, referred to internally as the 'relaunch', was subsequently commissioned in June 2013 to significantly overhaul the new system and address the identified shortcomings. The relaunch was tasked with a full re-evaluation of the system requirements, identification of the new current system's issues, and improvement of the application's functionality. As part of the relaunch, formalized roles and responsibilities were established, including a document outlining the TechRecruit Steering Committee and Core Project Team. The Steering Committee consisted of five TechRecruit senior executives and three senior managers from the Super-CRM vendor. The Core Project Team consisted of a TechRecruit Program Manager, four TechRecruit Project Managers, one Super-CRM vendor Project Manager, and 21 TechRecruit business and IT representatives.

As we will illustrate in the next section, despite the traditional approach undertaken during the first project phase, the relaunch adopted a hybrid collection of traditional techniques and agile techniques based on the Scrum methodology. This included the use of story cards, four-week sprints, stand-up meetings, and planning poker. While all these techniques required frequent face-to-face interactions and an increasingly iterative nature, a range of traditional techniques from the first phase also remained in place. The TechRecruit Program Manager was a Certified ScrumMaster and the company had previously used Scrum for technology infrastructure projects, but the remainder of the project team had very limited experience with agile methods at the outset of the project. The relaunch consisted of a planning phase (e.g. user interviews and scope setting), iterative code development, system testing and validation, system deployment, and user training. The relaunched system was rolled out successfully with greatly improved performance and functionality during February 2014.

Our rationale for choosing TechRecruit to participate in this research is based on the position that a wide range of companies are in a similar position of having deep experience with traditional development techniques, but are still developing their agile capabilities. As agile is increasingly used in such companies, a variety of challenges are introduced, but no clear solutions have yet been outlined. Indeed, recent practitioner reports suggest that 43% of organizations have less than 2 years of experience with agile and 33% of agile adoptions remain in an 'early' phase

(VersionOne, 2016). Furthermore, the three leading causes of failed agile projects are noted to be a company culture that is at odds with agile values (46%), a lack of experience with agile methods (41%), a lack of management support (38%) (VersionOne, 2016). On this basis, we argue that a) TechRecruit is representative of many other companies in their limited experience with agile; b) TechRecruit experienced many of the same agile challenges faced by other companies; and c) the successes experienced by TechRecruit in navigating these challenges will be of great interest to other, similar companies.

Findings and Analysis

Our case revolves around the sequence of events noted above and aims to unveil the changes in knowledge sharing processes that originated with traditional development techniques used in the first phase, but shifted to a partial use of agile techniques during the second phase. In presenting the results of our study, we draw on the Chau et al. (2003) framework (Table 1) and focus on three *patterns* of knowledge sharing processes that emerged between the first and second project phase. First, we discuss the knowledge sharing processes that continued to use traditional techniques in both the first and second phases. Second, we discuss the knowledge sharing processes that combined traditional techniques and agile techniques into a hybrid approach. Third, we discuss knowledge sharing processes that drew on primarily agile techniques[1]. For each knowledge sharing process pattern, we also identify key *factors* that led TechRecruit to stick with traditional techniques, adopt a hybrid collection of ISD techniques, or switch to primarily agile techniques.

Knowledge Sharing Processes that Continued to Use Traditional Techniques in the Second Project Phase

Our analysis showed that both the ‘use of documentation’ and ‘training’ knowledge sharing processes were characterized by the use of traditional development techniques in both the first and second phases of the project.

Use of documentation: The use of documentation during the first project phase consisted of a wide range of process workflows, project plans, and resource estimations. This was in line with traditional development techniques. As the project transitioned into the relaunch, TechRecruit’s relationship with the software vendor became increasingly tenuous and clear documentation and guidelines were viewed as a mechanism to ensure they delivered on their promises. Management expressed concerns that any shift towards a more agile-oriented approach of minimizing project documentation would prove to be too risky, as the vendor couldn’t be trusted to work without explicit guidelines. Documentation was viewed by TechRecruit as a means to reinforce and clearly establish

[1] Additional examples of data associated with each knowledge sharing process that supplements the representative examples below are noted in Appendix C.

expectations and ground rules for the vendor. This familiarity with the existing, traditional techniques was essentially used as a failsafe against further performance issues, since TechRecruit management believed that the use of formal documentation allowed them to maintain a degree of project control by requiring the vendor to explicitly state what tasks they were responsible for performing. As the CIO told us:

> *We decided that we wanted to force [the vendor's] hand on a technical strategy document that they would actually sign up to. And the reason behind that was that we had a lot of quality issues with them. So as part of the technical strategy document, which actually I wrote, we required them to go through and specify how they would do assessment level testing.* – CIO

The key factor that led Tech Recruit to continue using traditional techniques for the 'use of documentation' knowledge sharing process rests on the lack of trust that TechRecruit had with the application vendor. Therefore, written documentation played a relevant role in guiding the vendor's activities. It is possible that if the project been fully insourced or had management trusted the vendor more, the traditional orientation towards documentation may have been relaxed in favor of a more agile approach.

Training: From a training perspective, a highly structured, lecture-based, classroom training approach was used in the first phase of the project. This approach was viewed as a valuable technique by the recruiters and trainers, largely due to the wide range of technical competencies of end users and lack of familiarity with the new application. Since the new system was markedly different than the legacy system, users were frequently confused and uncertain about the new functionality. Classroom-based formal training allowed users to receive structured guidance and gain confidence in using the system. This approach was continued during the relaunch, as additional formal, hands-on training sessions were conducted, based on materials derived from the first implementation. In particular, following the technical challenges of the failed first phase, trainers viewed the formal training in the second phase as continuing to be important. Users had struggled with the design and navigation of the initial system and required additional guidance to master the relaunched system. The decision to continue with a structured training approach was largely driven by the positive feedback that was received related to the confidence that the training was able to cultivate in employees during the initial implementation. This was highlighted by a training specialist:

> *People lacked confidence. Technology is a scary thing for a lot of people…when you introduce a new system, you will meet a lack of confidence. The only way to win with this lack of confidence is to get people to successfully work with the system. Like a videogame, at the beginning aliens win and you are blown away… So the first part has great importance, because it is really difficult… Then, you are okay… and users have to take confidence with the system, because if this doesn't happen, they don't accept the system.* - Application Training Specialist

The key factor contributing to the 'training' characteristics during the second project phase is associated with management's recognition that the technical capabilities of users remained a concern and shifting towards a more informal approach to training would leave many users uncomfortable with the system functionality. Formal training had been viewed as effective during the first phase of the project, even if the system itself was viewed as a failure. Repeating this valuable activity was viewed as the most prudent approach.

Knowledge Sharing Processes that Shifted to Hybrid Techniques in the Second Project Phase

Agile techniques were introduced alongside traditional techniques for three knowledge sharing processes during the project relaunch: 'competence management', 'trust and care', and 'team composition'.

Competence management: One of the most significant issues for TechRecruit management during the first project phase was the realization that the vendor was understaffed and unresponsive. In addition to maintaining formal documentation, as noted in the section above, the relaunch also utilized more careful monitoring of project issues and more frequent meetings with the vendor. Although TechRecruit didn't employ a fully agile approach, techniques such as stand-up meetings were introduced and were held twice weekly in order to improve the IT Project Manager's ability to address issues and find solutions. This increased level of communication and more intensive verification of the vendor's activities allowed TechRecruit management to more effectively monitor project issues and concerns. This 'middle ground' between a traditional and agile approach was indicative of the hybrid strategy that was emerging on the project, which sought to leverage the company's existing capabilities, while supplementing the ISD team with selected and focused agile techniques where they felt it could aid performance. In effect, management didn't believe that the 'competence management' knowledge sharing process was completely broken (in which case, it would require a more extensive overhaul). Rather, management believed that the process could be tweaked with the addition of some agile techniques to surgically address the underperforming elements of the process. In line with this view, IT project manager noted that:

> *The development team had their own agile approach, but my relationship with their project manager was onboard with what our agile approach is. So, we had to align our tasks knowing that certain development pieces were going to be completed as part of their sprints. It was definitely in parallel and their project manager was knowledgeable and onboard with what we were doing. So they knew that when we did Scrum planning, I would talk to him and say, 'we will tackle this in this sprint'. I would communicate that to the project manager. 'Is this doable?', 'do you think that this is going to get done on this sprint?'.* – IT Project Manager

The key factor leading to the shift from traditional techniques to hybrid techniques in the 'competence management' knowledge sharing process is related to the fact that gaining insights into the daily status of the project restricted management's ability to monitor vendor activities. By adding an agile component oriented around communication and interaction, this challenge could be minimized by more frequently collected project status updates.

Trust and care: Although a hard deadline was established for the first phase go-live, several interviewees disagreed on whether the system was actually ready to be implemented or if the date should have been pushed back. This disconnect appears to have created a level of distrust between TechRecruit management and staff. In response, during the relaunch, tasks were re-prioritized and the level of effort was estimated using a planning poker approach and team members were given autonomy to sign up for their choice of tasks. This was viewed as helping to facilitate a shared understanding of project scope, priorities, and expectations amongst both managers and staff. Similar to the competence management changes noted above, this partial shift towards agile was an important initiative for management to demonstrate that they were recognizing employee concerns. Rather than a wholesale transition of trust and autonomy to lower level employees, as would have been required under an agile approach, TechRecruit management more fully engaged staff during the relaunch, but maintained a degree of control over the final decision making. A business analyst elaborated this point:

> *Management did not get involved as much as they should have when we [first] implemented…I did not have a business owner when we implemented this application and I fought for six months afterwards to get business owners. And I now finally have a business owner for sales and a business owner for recruiting and marketing owns the marketing piece, so now I have people who know the direction of the company and that can help me develop the app in the way it needs to go for the business, not for individual users.* – Business Analyst

The key factor causing the 'trust and care' knowledge sharing process to employ hybrid techniques lies in the (partial) lack of trust that emerged during the first phase of the project; disagreements between management and staff on preparedness for the go-live had strained the trust between the two groups. In turn, by employing a more decentralized approach to project oversight, management was able to gain an enhanced perspective into the project because staff members felt more engaged and trusted by management.

Team composition: The composition of the development team during the first project phase was restricted to a small group of organizational representatives who participated as 'pilot team' members during requirements gathering and testing. Although this structure provided useful feedback during the project, interviewees suggested that it didn't sufficiently represent the opinions and perspectives of the wider organization. The relaunch restructured the project team into a more decentralized model, such that team leaders were set up to oversee staffing deliverables, finance deliverables, and IT deliverables, while a ScrumMaster oversaw the project activities and backlog. This group ran as an agile team, but each team leader independently managed the activities related to their deliverable area. In doing so, a broader pool of employees were able to contribute their viewpoints into project progress and system functionality, while still maintaining a small number of key decision makers. This 'middle ground' allowed management to better draw on the skills and knowledge of a wider collection of employees, without sacrificing the centralized decision making that they viewed as important to project governance. To this end, the CIO recalled issues related to the first phase and pointed out that:

> *I was the project manager for IT…so let's say one of the IT deliverables would be that we had to create 15 laptops with a localized email environment and other configurations on the laptop specific to training purposes. So what I would then do is I would negotiate with my infrastructure team to get them to agree to a deliverable where I would actually set the expectations around requirements and what needed to get done. Then I would basically come back to [the Scrum Master] and say, 'Okay, I'm committing for this sprint to get this effort done and it will be done within this four-week period of time'. So although the team who was acting on it wasn't really part of the agile [team], I was part of the agile team and then I worked to extend myself and then to basically commit to doing what I needed to get done.* – CIO

The perception that users were insufficiently involved in the first phase represented a key factor leading to increasing the breadth of participation amongst the users and business stakeholders during the project relaunch. Therefore, the 'team composition' knowledge sharing process was modified to employ both traditional and agile techniques in the second project phase.

Knowledge Sharing Processes that that Shifted to Agile Techniques in the Second Project Phase

Our data analysis indicated that the knowledge sharing processes of 'requirements and domain knowledge', 'continuous learning', and 'knowledge repositories' were supported with predominantly agile development techniques in the second project phase.

Requirements and domain knowledge: During the first project phase, the project team members demonstrated an adequate understanding of why the system was being implemented and what it was designed for. Although many of these requirements were met when the system initially went live, interviewees identified a variety of significant shortcomings, including missing features, poor performance, and issues with reliability. In an attempt to re-evaluate and prioritize what requirements were important for the relaunched system and how it could be more effectively designed, the second project phase transitioned away from formal ownership of particular deliverables and increasingly relied on cross-functional discussions, consistent with an agile approach. This shift diverges from the previous hybrid examples in that TechRecruit managers appeared more willing to adopt a largely new approach to address a significant weakness during the initial implementation. Based on the widespread displeasure that employees expressed in regard to requirements delivery, management recognized that a dramatically new approach was needed for the relaunch. However, the company also realized that they would need to carefully pick and choose the areas where agile techniques would be accepted by staff and could contribute immediately to project performance. To do so, management sought to bring together a variety of forward-thinking staff who could contribute to more effectively define the requirements for the relaunch. As the CIO told us, this inter-disciplinary approach to requirements and domain knowledge consisted of:

> *The essence [of agile development] is really the culture, the fit, getting people's head around that and you might find some people, who although they are technically very talented, they're just unable to come to grips with working and being willing to share and be willing to extend themselves…the idea about agile is that even though I'm a developer, I should be able to write functional tasks…if I'm a business development guy I should be able to do some UI [user interface] work. It might be that I'm not a user interface expert but now gaining some appreciation of what it takes to basically implement the user and integrate that with the business layer….So we've got that in terms of how to do that more effectively.* – CIO

The above is illustrative of the key factor enabling a transition of the 'requirements and domain knowledge' process from traditional techniques to agile techniques. The concerns around isolated decision making demanded a more collaborative, integrative approach. By selecting a new, forward-thinking collection of project participants with broad knowledge of how the business at TechRecruit works, management was able to do away with the siloed approach to requirements gathering and bring a new, integrative perspective to the project's second phase.

Continuous learning: This was another knowledge sharing process that shifted in the project's second phase to a predominantly agile approach. Due to the initial performance and reporting issues with the first project phase, users became increasingly familiar with the system functions through trial and error in the months following go-live. In order to better harness these insights, monthly sprint retrospectives were introduced for TechRecruit project team members during the project relaunch. Discussions included what went well, issues that occurred, and opportunities for improvement. Team members became increasingly comfortable with agile techniques, such as planning poker, where team members estimate the effort work required to complete a project task. This more responsive approach to addressing project issues was highly appealing to both staff and managers, as it allowed them to quickly respond and adapt to changing project characteristics. In part because of the perception that the initial implementation had been too slow to respond to signs of trouble, project team members were primed to embrace a novel approach to dynamically address project issues during the relaunch. This positive outlook increased during the project as staff recognized the incremental benefits to project performance. Interestingly, the Director of Application Management was able to describe this transition as emerging and 'bottom-up'. Although few staff had experience in short, iterative development sprints, they all found this new technique to be extremely effective to move ahead with the development while continuously checking the effectiveness of their work on the system. The following quote explains this evolution:

> *As the sprints progressed…I might not even have known where I was pulling the numbers from as far as estimation of hours and things like that. But as we went through the sprint and had the meetings weekly and then were following everything in here, I'm sure I was a lot more accurate in sprints three, four, five, six and seven [compared to] the first couple. So it got refined as each sprint went forward.* – Director, Application Management

The need to obtain dynamic, informal feedback (required by project stakeholders) represented the key factor leading the shift of the 'continuous learning' process to move from traditional to agile techniques. In fact, the users would have no longer accepted long implementation delays and, due to the unsuccessful rollout of the first phase they were willing to provide any helpful feedback, had the system worked for them quickly and effectively.

Knowledge repositories: During the first project phase, knowledge repositories including Microsoft SharePoint, were in place to store formal project documents, but interviewees indicated that usage of the repository declined as go-live approached. The repositories were viewed as containing documents that were applicable to the broader administration of the project, but provided little aid in the day-to-day challenges of participating on the project or using the system. In response, the relaunch employed a more extensive, traceable repository of project knowledge thorough the use of the lightweight agile tool LeanKit[2]. This included story cards and the project backlog, which allowed a real-time view of the project status for management and staff. The tool was extensively used throughout the relaunch, as it housed the relevant status of tasks, issues, and progress, while encouraging only a minimum of conventional project documents and plans. Rather than mandating use of the SharePoint tool (which would correspond with a more traditional approach because of its reliance on lengthy documentation for all aspects of the project), TechRecruit management's decision to shift to LeanKit demonstrated a shift towards a more autonomous, self-directed approach that was consistent with agile. That is, the new tool became a fundamental knowledge sharing mechanism that enabled staff to perform their jobs more effectively, in contrast to the previous tool, which was viewed as only as a bureaucratic and administrative inconvenience by staff. As the IT project manager pointed out:

> *We looked into a couple software options to help streamline that process and…I came up with LeanKit because they had the whiteboard with the sticky notes, you could assign tasks to people, I could schedule certain things, and we had pretty good analytics to see tasks being completed.*

[2] https://leankit.com

What's reasonable for how many hours are allocated to each person and then I could track things that came up during our sprints that needed...I'd put them to a parking lot as they called it, because it needed to be vetted out further outside of our daily scrum. - IT Project Manager

The key factor leading to the adoption of LeanKit was the lack of user engagement with traditional knowledge repository (i.e., SharePoint) in the first phase. In turn, management started to consider a more useful repository that would better align with the increasingly dynamic, collaborative development style of the second project phase.

Table 2 provides a summary of all key factors leading to the various configuration (or reconfiguration) of ISD techniques that we discussed above.

Knowledge Sharing *Patterns*		***Factors* Leading to Second Phase Development Characteristics**	**Corresponding Knowledge Sharing Process**
First Phase Development Techniques	**Second Phase (Relaunch) Development Techniques**		
Traditional	Traditional	Lack of vendor trust	Documentation
		First phase development techniques were found to be effective	Training
Traditional	Hybrid	Limited insights into project status	Competence management
		Management-staff disconnect	Trust and care
		Insufficient user participation	Team composition
Traditional	Agile	Isolated decision making	Requirements and domain knowledge
		Need for more dynamic, informal feedback	Continuous learning
		Lack of user engagement with current tools	Knowledge repositories

Table 2. TechRecruit Development Techniques and Knowledge Sharing Processes

In summary, our findings suggest that a series of key factors within the ISD project led to the adoption of selected agile techniques, which in turn influenced the knowledge sharing processes in the subsequent phase of the project. Some of these factors were associated with the recognition that traditional ISD techniques were contributing to effective knowledge sharing in particular processes and should continue unchanged. However, for other knowledge sharing processes, we find that the factors associated with concerns around trust, user participation, decision making, and team member engagement led to the introduction of agile techniques. In some cases (e.g., team composition) this transition was slow and incremental, but for others (e.g., knowledge repositories) it was increasingly rapid and extensive. In the next section, we discuss the findings through the lens of ambidexterity theory and highlight the implications of our study for research and practice.

Discussion and Implications

In considering the results presented above, we reflected on the increased success of the second project phase, relative to the first phase. Although the use of selected agile development techniques provided TechRecruit with a novel approach to address the knowledge sharing shortcomings identified in the first phase of the project, interviewees did not indicate that it was only the introduction of agile that contributed to improved project performance or that adding even more agile techniques would contribute to further performance improvements. Instead, we found that the continued use of traditional development techniques in some knowledge sharing processes such as documentation and training, tangibly added value to the project during the relaunch. In other knowledge sharing process, such as team composition, a hybrid mix of agile and traditional techniques was viewed by interviewees as effective. In order to highlight the practical and theoretical insights that can be gleaned from our analysis, we considered the patterns of knowledge sharing noted within the empirical data as the project shifted from traditional to agile, specifically when viewed through the lens of ambidexterity theory.

The Emergence of Ambidextrous Capabilities in Hybrid Projects

During the second project phase, evidence of a shift towards the pursuit of ambidexterity became increasingly clear in that capabilities were cultivated across both traditional and agile approaches. As noted in our results, two out of the eight knowledge sharing processes remained primarily traditional in nature, but three shifted to hybrid and three progressed to become primarily agile (see Figure 2). This highlights the unique configuration of agile and traditional techniques that contributed, at least in part, to the positive results of the second project phase. We would expect a degree of this improvement to be attributable due to learning effects (e.g., the team's experience with training in the first phase could enable them to do an even better job in the second phase); however, six of eight knowledge sharing processes included a substantive change to new (or partly new) ISD techniques that the team was unfamiliar with. Although we recognize that each organization's ISD approach is unique and a degree of overlap may exist in some circumstances (e.g., exploitation of knowledge with agile), our findings suggest that TechRecruit developed the ambidextrous capability to simultaneously balance both knowledge exploration (i.e., traditional techniques) and knowledge exploration (i.e., agile techniques). In the following section, we identify a series of ambidexterity-oriented *themes* that emerged from our findings through the use of temporal bracketing (discussed in the Methodology). By comparing the empirical data in the first and second project phases, we were able to further clarify the role of ambidextrous capabilities in the TechRecruit project.

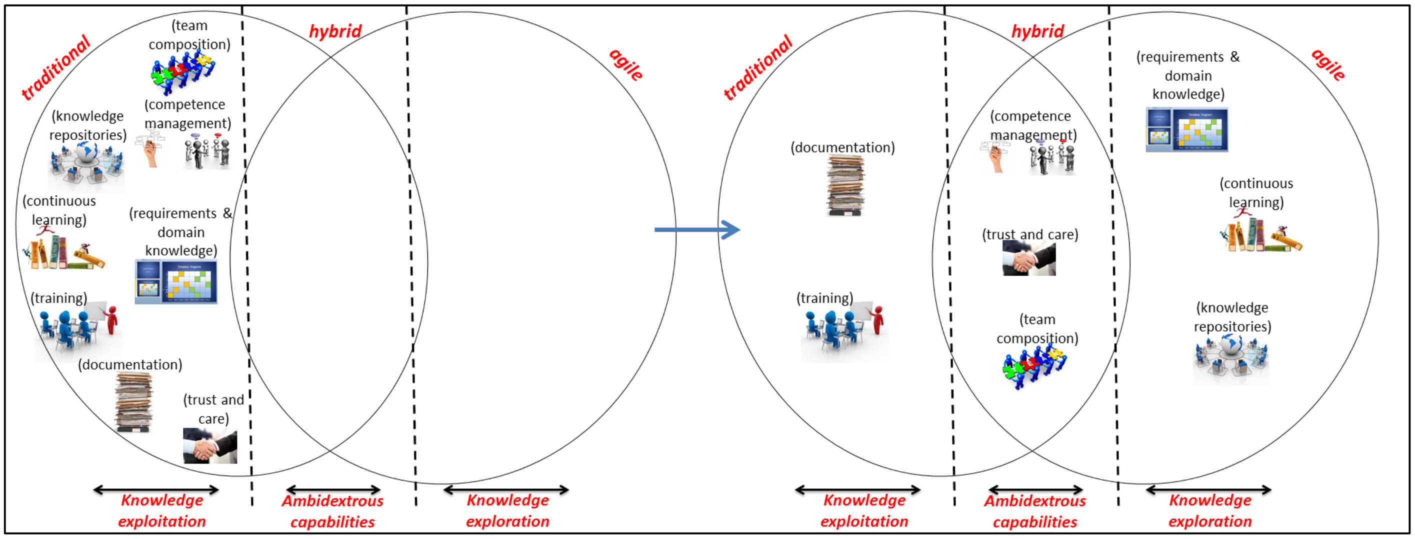

Figure 2. Knowledge Sharing Processes at TechRecruit

When considering the nature of the knowledge sharing processes that either remained traditional, shifted to hybrid, or transitioned to agile, we noted a set of associated characteristics. First, the knowledge sharing processes that remained traditional (documentation and training) are primarily oriented around explicit knowledge. That is, the development techniques that TechRecruit was least willing to abandon were associated with written documents, formal policies, and classroom training sessions that management and staff viewed as important. For companies that are wary of beginning down the path towards agile, maintaining these types of development techniques may provide a sense of familiarity and stability.

Second, the knowledge sharing processes that shifted to hybrid (competence management, trust and care, team composition) focus primarily around team communications. For example, rather than relying only on formal status reports (a traditional technique), the project relaunch *also* engaged stakeholders in informal communication to collect improved insights (an agile technique). Similarly, in order to better facilitate user involvement during the relaunch, TechRecruit management adjusted the team structure to a more decentralized model that allowed a broader range of employees to contribute their viewpoints. For companies that struggle with team communications on their projects, but are reluctant to go to a fully agile configuration with open workspaces and cross-functional teams, employing such hybrid techniques could be a promising alternative. The emergence of team communication as a knowledge sharing category that reflects a hybrid approach (e.g., with some traditional as well some agile components) fits with the idea of combining knowledge exploration and exploitation aspects within the same knowledge sharing process. Indeed, prior literature has provided evidence that team communication can be either managed through the use of explicit knowledge (e.g., Grant, 1998), tacit knowledge (e.g., Leonard and Sensiper, 1998), or both (e.g., Nonaka, 1994, Nonaka and Konno, 1998).

Finally, the knowledge sharing processes that transitioned to agile (requirements and domain knowledge, continuous learning, knowledge repositories) concentrate on tacit knowledge. In contrast to the explicit knowledge that is more straightforward to document and share, tacit knowledge is created and shared through experience and social interactions (Nonaka, 1994). This reflects the particular empirical settings examined, where despite the project manager's knowledgeability on agile implementations, the whole staff had to 'learn by doing'. For example, TechRecruit staff shared tacit knowledge through techniques such as sprint retrospectives, where open discussions were held by team members regarding what went well in the previous sprint and what could be improved in the next sprint. For companies seeking to transform their capability to cultivate and share tacit knowledge, these techniques represent a notable improvement compared to traditional techniques that focus primarily on explicit knowledge.

Practitioner Guidance and Contributions

We highlight these noted patterns in Table 3[3], alongside supplementary guidance for practitioners. Specifically, we suggest that for organizations experienced with traditional approaches that are considering transitioning to a hybrid approach, one possible tactic is to orient the transition around the techniques associated with different knowledge sharing processes. Although we recognize that this may be one of a series of possible approaches, it draws on the success experienced by TechRecruit and may be beneficial to other organizations in a similar situation, making our findings transferrable across contexts. Each row in Table 3 highlights the proposed link between a knowledge category (e.g. explicit knowledge), a sharing pattern (e.g., remains traditional), and sample ISD techniques that correspond with the process (e.g., the use of cross-functional teams).

Our analysis also uncovered four, high level *themes* that influenced the choices made by TechRecruit management in navigating their transition from traditional to hybrid ('Influencing Themes' section in Table 3). In attempting to generalize the circumstances at TechRecruit to other company situations, we have annotated the extent that they correspond with each knowledge sharing process and how this might apply elsewhere. We provide a more detailed assessment of each of the themes in the discussion below, including how the insights relate to past theorizing, possible applications in future research, and useful insights for practitioners.

3 Along with the idea that team communication is neither 'always formal' (traditional) nor 'always informal' (agile), agile approaches incorporate a relatively minor component of explicit knowledge and traditional approaches a relatively minor component of tacit knowledge. However, for analytical purposes, in Table 1 we made specific associations between types of knowledge / team communications and the traditional and agile aspects.

Knowledge Sharing Category	Knowledge Sharing *Pattern*	Corresponding Development Techniques	Influencing Themes			
			Ambidextrous Capability Development Over Time	**Control Issues and the Need to Perform**	**Size and Informality**	**Role of Informal Knowledge Management Systems**
Explicit knowledge	Remains Traditional	• Formal requirements documentation and project plans • Formal training sessions			X	
Team communications	Agile-Traditional Hybrid	• Mix of formal reports and informal stakeholder communication • Moderate reliance on team trust • Moderately cross-functional teams	X	X	X	
Tacit knowledge	Transitions to Agile	• Lightweight, informal tools • Pair programming • Project retrospectives • User participation • User stories		X	X	X

Table 3. Knowledge Category Links to Development Techniques and Influencing Themes

Theme #1 - Ambidextrous capability development over time: Our data suggest that the structural and cultural challenges that companies face when shifting from a purely traditional to a purely agile approach may be eased with a hybrid approach that emerges over time. To this end, management selected techniques that were easiest to implement, were perceived to be acceptable to staff (who were not experienced with agile development), and addressed the problem areas within the existing traditional approach. For example, in the first phase of the CRM implementation, many required software functions were delayed due to staff shortages or were not sufficiently tested prior to going live. By adopting four-week sprints and timeboxing in the relaunch, alongside current traditional techniques, the new agile techniques were perceived as a solution to address the first phase problems, including a lack of vendor trust and the management-staff disconnect (see Table 2), but were not so revolutionary as to create staff resistance that might have occurred had agile techniques been used exclusively. In other words, the demand to explore new knowledge (via agile techniques), while also exploiting existing knowledge (via traditional techniques), rose gradually. This allowed team members to become more comfortable with the transition, which is illustrative of past views on how ambidextrous capabilities develop (Turner et al., 2013). Indeed, past studies frame ambidexterity as a dynamic capability (O'Reilly and Tushman, 2008, O'Reilly and Tushman, 2011), which emerges over time through a variety of high level organizational mechanisms such as social integration (at the management level) and boundary spanning (at the operations level) (Jansen et al., 2009). Our study takes these insights further. Building on prior research we were able to provide more granular evidence of how ambidextrous capabilities emerge in the context of a hybrid project, with a specific focus on knowledge sharing.

As noted in Table 3, TechRecruit's shift towards ambidexterity was oriented around the adoption of new ISD techniques that extended their capabilities related to team communications and tacit knowledge sharing. Since they viewed their traditional development techniques as effectively exploiting the sharing of explicit knowledge, they decided to remain static in that area. Our data provide unique insights into the view that incrementally introducing staff to new development techniques in order to generate their buy-in can encourage a smooth, albeit slower, transition to innovative practices, thus leading to the development of ambidextrous capabilities: the team's ability to explore (novel) agile techniques, while simultaneously exploiting existing traditional techniques.

Theme #2 - Control issues that are influenced by the 'need' to perform: One of the key knowledge sharing considerations for companies pursuing an ambidextrous ISD approach is the shifting balance of control from management to project team members that is associated with agile techniques (Nerur et al., 2005, Nerur and Balijepally, 2007). Such a shift could be in conflict with cultural norms and generate resistance from management (Walsh and Seward, 1990, McDermott and O'Dell, 2001). However, in situations where the traditional approach is producing less than ideal results – as it was at TechRecruit during the first project phase – the transition towards a hybrid approach may be made more easily. We found little evidence to support the assertion that managers viewed agile as a threat to their control of knowledge on the project. Instead, as noted in our results, management employed agile techniques such as the LeanKit tool, story cards, and planning poker in order to shift more responsibility to the lower levels as a means to generate increased engagement and collaboration.

Past research, such as McAvoy and Butler (2009) highlight potential team-level challenges related to decision-making processes and indicate that a conscious application of the devil's advocate approach by the project manager might be helpful, especially with empowered, cohesive teams. However, at TechRecruit, the project team viewed the introduction of agile techniques as an innovative means to create and share knowledge that would benefit the project and the company. This is in relative contrast with some of the agile literature that discusses challenges associated with the shift of power from management to the team (Williams and Cockburn, 2003, Nerur et al., 2005).

However, our findings suggest that a one of the reasons why this change was made possible relates to the cohesiveness of the team that understood successful project outcomes could be achieved by integrating agile development techniques with traditional techniques. As noted by McAvoy and Butler (2009), a cohesive team does not always lead to successful agile projects; however, TechRecruit's shared motivation for project success represented an asset for developing the ability to both explore and exploit knowledge related to the project. Therefore, by building on prior research we provide a novel insight in the ISD literature that relates to a firm's ability to manage power shifts using past failures and lessons learned as way to let organizational actors understand the relevance to address collective efforts into knowledge sharing processes that will positively affect ISD outcomes.

Theme #3 - Size and informality as characteristics promoting ambidexterity: Vinekar et al. (2006) outline a model of traditional and agile co-existence that draws on the concept of ambidexterity. In their example, relatively large organizations employ two or more independent sub-units, some with high exploitive ability that adopt a traditional approach and others with high explorative ability that adopt an agile approach. From a practical perspective, this option avoids at least some of the trade-offs that arise within a hybrid approach, such as the need for members of the same team to rely on both explicit, written knowledge, as well more tacit, informal knowledge from social interactions. However, integrating the outcomes of different business units requires integration mechanisms, such as the use of product/process managers who can act as boundary spanners between different units (Lubatkin et al., 2006). Midsized companies like TechRecruit typically do not have the size and resources to employ two independent ISD teams and associated line management to avoid a silo effect between business units (e.g., between those units pursuing knowledge exploration and those pursuing exploitation). Therefore, a company's size and resource constraints might drive hybrid solutions to require a certain degree of improvisation and aid in overcoming the rigidity that characterizes traditional approaches (Ciborra, 2000, Verjans, 2005).

Moreover, midsized companies are generally more flexible, exhibit a greater level of informality and tend to be challenged with less complexity in adopting agile that larger organizations (Lindvall et al., 2004, Damanpour, 1992, Haveman, 1993, Jansen et al., 2009). For instance, from our field observations it emerged that most project team members knew each other quite well - even if they had different responsibilities and did not work exclusively on the same projects. Andriopolous and Lewis (2009) are among the few scholars who discuss links between organizational size and ambidexterity, yet they do not explicitly consider the emergence of ambidextrous capabilities within midsize firms. Although past research has already assessed that midsize companies are more likely to learn quickly by informal interactions (Nevis et al., 2009), neither the ambidextrous literature nor the ISD literature have related a company's size with the ability to develop ambidextrous capability and thus being able to pursue hybrid projects. Our findings represent a theoretical contribution and a promising area of future research.

Theme #4 - Relevance of informal knowledge management systems (KMS): The LeanKit collaboration tool represents an IT artifact that was able to replace an unused knowledge repository and promote team member interaction. Past research highlights key elements such as validation that can discourage the use of knowledge repositories (Fadel and Durcikova, 2014). This illustrates the shift from a traditional repository approach, where 'static' knowledge is taken for granted and is used and applied mechanically (Wagner and Newell, 2004) to a more networked approach (Bresnen et al., 2003, Newell, 2015) where knowledge is continuously shaped by users from different departments, which is consistent with an agile approach. For TechRecruit, LeanKit provided an agile-oriented repository that was flexible enough to remain compatible with the other hybrid and traditional knowledge sharing processes being employed on the project (i.e., supporting the use of 'minimal' documentation, while encouraging team communications). Interestingly, while past research has highlighted how IT artifacts might help share knowledge 'dynamically' (Galliers and Newell, 2003, Alavi and Leidner, 2001a, Newell et al., 2009), there are very few agile-related studies that bring this aspect to the fore (one exception is Joshi et al. 2007). More importantly, in this theoretical contribution we point to the relevance of knowledge management systems not just to support agile projects, but also to deal with transitions from traditional to hybrid, where some knowledge still lies in static repositories (e.g., SharePoint for TechRecruit) and other knowledge needs to be dynamically shared. To this end, one possible direction for future research concerns how far IT artifacts in ISD contexts promote 'learning by doing' practices (e.g., how to become partially agile, as per our fieldwork) by supporting people's existing and somewhat 'static' familiarity with traditional approaches, while fostering learning, communication, and collaboration processes. While the role of IT artifacts in learning process is not new to the IS literature (e.g., Alavi and Leidner, 2001b, Kane and Alavi, 2007, Sambamurthy et al., 2003), again here we build on, and integrate two streams of research that we believe are relevant in order to gain a better understanding of how traditional-to-hybrid transitions occur when staff has little of no experience with agile. While Table 4 synthesizes the above, the next concluding section outlines limitations and future avenues of research.

Influencing theme	Theoretical contribution to the MIS literature	Contribution to ISD practitioners
Ambidextrous capability development over time	Shifts in knowledge sharing processes is an organizational mechanism (not considered by prior literature) that promotes the development of ambidextrous capabilities.	Hybrid approaches can be pursued by focusing on knowledge sharing aspects. Particularly when teams are not agile savvy, balancing these processes might make the transition smoother while promoting 'learning by doing'.
Control issues and the need to perform	Decentralization of power/control might be achieved by providing organizational actors with shared values associated with project success. This insights has been a somewhat studied topic in the management literature but has never been systematically analyzed in ISD contexts. To this end we question Nerur et al. (2005) and provide a counterintuitive insight which opens up the opportunity to further examine the management of power issues in ISD projects.	Project managers should focus on engaging and committing staff in ways that go beyond individual performance associated with financial compensation. Instead, a holistic and long-term view of the relationship between project success (collective achievement) and individual performance should be promoted.
Size and informality	Organizational size affects the ability to be ambidextrous. Here we build on – and extend – the work of Michael Tushman and Charles O'Reilly (e.g., Tushman and O'Reilly, 1996, O'Reilly and Tushman, 2008).	Small and medium sized companies should leverage their size and informality to explore new ISD strategies (e.g., hybrid), even if the various teams are not specifically trained in novel approaches, as they will have more chances to learn by doing. However, management should remember that staff engagement and commitment will be key to have people participating in 'hands-on' learning.
Role of informal knowledge management systems	IT artifacts can support learning processes (e.g., referred to new approaches) in ISD contexts.	The choice of knowledge management tools is key in ISD projects. In particular, flexible tools might help transitions from traditional to hybrid approach because they allow the survival of existing (traditional) knowledge sharing processes and the exploration of new, more flexible and dynamic processes.

Table 4. Theoretical and Practitioner Contributions

Conclusions

As with any study, our work has limitations, as well as opportunities for further research. First, our focus on the knowledge sharing elements of TechRecruit's development project was shaped by the framework proposed by Chau et al. (2003). This focus, in our opinion, provides the benefit of creating in-depth insights, but creates the inevitable shortcoming of leaving other, broader issues unexplored. Future research could build on this foundation by expanding beyond our knowledge sharing orientation to increasingly consider other aspects of ISD, such as testing and maintenance. Second, our study focuses on one project at one company. By considering a wider range of companies and projects, future research could continue to refine the knowledge sharing and ISD themes that we note above. Third, we studied an ISD project that worked to significantly customize an existing vendor product. Future research could consider if different results might exist in projects that employ other development approaches, such as a system developed completely in-house. Finally, the empirical data collected for this study was sourced exclusively from TechRecruit employees. Although some of this data include accounts of communications with the vendor representatives, we were unable to conduct first-hand interviews with vendor employees. Future research may be able to further build on our results by including the perspectives of a broader range of project stakeholders.

Our study makes important contributions to theory and practice. First, we contribute to theory by elaborating on the notion of ambidexterity in hybrid systems development contexts. In doing so, we build on prior ISD research to propose that ambidextrous capabilities develop over time and are influenced by a series of key factors (e.g., trust, project transparency). Our results suggest that these capabilities form common themes, such as the need to perform, organizational size, and the use of informal knowledge management tools. These theoretical insights are novel to the ISD domain and we believe they deserve further investigation with complementary methods and techniques, which can enhance the generalizability of our findings. Second, we contribute to practice by examining the unique knowledge sharing benefits that can result from a hybrid systems development approach, as well as how organizations determine when to continue using traditional development techniques and when to replace them with agile development techniques. By understanding TechRecruit's path from a traditional approach to a hybrid agile-traditional approach, this study aids managers overseeing hybrid projects in context where the ISD team is not yet experienced with agile. We highlight the opportunities to transition to agile techniques in situations where team communications and tacit knowledge sharing can be improved, while being cautious with transitioning away from traditional techniques that relate to explicit knowledge sharing. Through an increasing awareness of the factors that drive the transition of traditional ISD techniques to agile techniques, particularly in situations where agile experience is limited, managers can more effectively attempt to achieve ambidextrous systems development capabilities and enhance ISD project performance.

References

AALTONEN, A. & TEMPINI, N. 2014. Everything counts in large amounts: a critical realist case study on data-based production. *Journal of Information Technology,* 29**,** 97-110.

ALAVI, M. & LEIDNER, D. E. 2001a. Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS Quarterly,* 25**,** 107-136.

ALAVI, M. & LEIDNER, D. E. 2001b. Research commentary: Technology-mediated learning—A call for greater depth and breadth of research. *Information Systems Research,* 12**,** 1-10.

ALZOUBI, Y. I., GILL, A. Q. & AL-ANI, A. 2016. Empirical studies of geographically distributed agile development communication challenges: A systematic review. *Information & Management,* 53**,** 22-37.

ANDRIOPOULOS, C. & LEWIS, M. W. 2009. Exploitation-Exploration Tensions and Organizational Ambidexterity: Managing Paradoxes of Innovation. *Organization Science,* 20**,** 696 - 717.

BALIJEPALLY, V., MAHAPATRA, R. & NERUR, S. 2006. Assessing Personality Profiles of Software Developers in Agile Development Teams. *Communications of the Association for Information Systems,* 18**,** 55-75.

BALIJEPALLY, V., MAHAPATRA, R., NERUR, S. & PRICE, K. H. 2009. Are Two Heads Better than One for Software Development? The Productivity Paradox of Pair Programming. *MIS Quarterly,* 33**,** 91-118.

BATRA, D. 2009. Modified Agile Practices for Outsourced Software Projects. *Communications of the ACM,* 52**,** 143-148.

BECK, K., BEEDLE, M., BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J. & MARICK, B. 2001. *Manifesto for Agile Software Development* [Online]. Available: http://agilemanifesto.org.

BENBASAT, I., GOLDSTEIN, D. K. & MEAD, M. 1987. The Case Research Strategy in Studies of Information Systems. *MIS Quarterly,* 11**,** 369-386.

BOEHM, B. 1976. Software Engineering. *IEEE Transactions on Computers,* C-25**,** 1226-1241.

BOEHM, B. & TURNER, R. 2003. Using Risk to Balance Agile and Plan-Driven Methods. *Computer,* 36**,** 57-66.

BOEHM, B. & TURNER, R. 2004. Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods. *Proceedings of the 26th International Conference on Software Engineering.* Edinburgh, Scotland.

BOEHM, B. & TURNER, R. 2005. Management challenges to implementing Agile Processes in traditional development organizations. *IEEE Software,* 22**,** 30-39.

BOSE, I., PAL, R. & YE, A. 2008. ERP and SCM systems integration: The case of a valve manufacturer in China. *Information & Management,* 45**,** 233-241.

BRESNEN, M., EDELMAND, L., NEWELL, S., SCARBROUGH, H. & SWAN, J. 2003. Social Practices And The Management Of Knowledge In Project Environments. *International Journal of Project Management,* 21**,** 157-166.

CABRAL, A. Y., RIBEIRO, M. B., LEMKE, A. P., SILVA, M. T., CRISTAL, M. & FRANCO., C. 2009. A Case Study of Knowledge Management Usage in Agile Software Projects. *Lecture Notes in Business Information Processing,* 24**,** 627-638.

CAO, L., MOHAN, K., XU, P. & RAMESH, B. 2009a. A Framework for Adapting Agile Development Methodologies. *European Journal of Information Systems,* 18**,** 332-343.

CAO, Q., GEDAJLOVIC, E. & ZHANG, H. 2009b. Unpacking Organizational Ambidexterity: Dimensions, Contingencies, and Synergistic Effects. *Organization Science,* 20**,** 781-796.

CHAN, F. K. Y. & THONG, J. Y. L. 2009. Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems,* 46**,** 803-814.

CHAU, T., MAURER, F. & MELNIK, G. Knowledge Sharing: Agile Methods vs. Tayloristic Methods. Twelfth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.

CIBORRA, C. U. 2000. *From Control to Drift: The Dynamics of Corporate Information Infrastructures,* Oxford, Oxford University Press.

CONBOY, K., COYLE, S., WANG, X. & PIKKARAINEN, M. 2011. People over Process: Key Challenges in Agile Development. *IEEE Software,* 28**,** 48-57.

CRAM, W. A. & BROHMAN, M. K. 2013. Controlling Systems Development: A New Typology for an Evolving Field. *Information Systems Journal,* 23**,** 137-154.

CRAM, W. A. & NEWELL, S. 2016. Mindful revolution or mindless trend? Examining agile development as a management fashion. *European Journal of Information Systems,* 25**,** 154-169.

CRAWFORD, B., CASTRO, C. & MONFROY, E. 2006. Knowledge Management in Different Software Development Approaches. *Advances in Information Systems.*

DAMANPOUR, F. 1992. Organizational Size and Innovation. *Organization Studies,* 13**,** 375-402.

DE CESARE, S., LYCETT, M., MACREDIE, R. D., PATEL, C. & PAUL, R. 2010. Examining Perceptions of Agility in Software Development Practice. *Communications of the ACM,* 53**,** 126-130.

DINGSØYR, T., NERUR, S., BALIJEPALLY, V. & MOE, N. B. 2012. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software,* 85**,** 1213-1221.

DURCIKOVA, A., FADEL, K., BUTLER, B. & GALLETTA, D. 2011. Knowledge Exploration and Exploitation: The impacts of psychological climate and knowledge management system access. *Information Systems Research,* 22**,** 855–866.

DYBA, T. & DINGSØYR, T. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology,* 50**,** 833-859.

ELBANNA, A. R. 2007. Implementing an integrated system in a socially dis-integrated enterprise: a critical view of ERP enabled integration. *Information Technology & People,* 20**,** 121-139.

FADEL, K. & DURCIKOVA, A. 2014. If it's fair, I'll share: The effect of perceived knowledge validation justice on contributions to an organizational knowledge repository. *Information & Management,* 51**,** 511-519.

FELDMAN, M. S. & ORLIKOWSKI, W. J. 2011. Theorizing Practice and Practicing Theory. *Organization Science,* 22**,** 1240-1253.

FITZGERALD, B., HARTNETT, G. & CONBOY, K. 2006. Customising Agile Methods to Software Practices at Intel Shannon. *European Journal of Information Systems,* 15**,** 200-213.

FRUHLING, A. & DE VREEDE, G.-J. 2006. Field Experiences with eXtreme Programming: Developing an Emergency Response System. *Journal of Management Information Systems,* 22**,** 39-68.

GALLIERS, R. D. & NEWELL, S. 2003. Back to the future: from knowledge management to the management of information and data. *Information Systems and e-Business Management,* 1**,** 5-13.

GHOBADI, S. 2015. What Drives Knowledge Sharing in Software Project Teams: A Review and Classification Framework. *Information & Management,* 52**,** 82-97.

GHOBADI, S. & MATHIASSEN, L. 2015. Perceived barriers to effective knowledge sharing in agile software teams. *Information Systems Journal,* Early view online.

GRANT, R. M. 1998. Toward a knowledge-based theory of the firm. *Strategic Management Journal,* 17**,** 109-122.

GUILLEMETTE, M. G., MIGNERAT, M. & PARÉ, G. 2017. The role of institutional work in the transformation of the IT function: A longitudinal case study in the healthcare sector. *Information & Management,* 54**,** 349-363.

HAVEMAN, H. A. 1993. Organizational Size and Change: Diversification in the Savings and Loan Industry after Deregulation. *Administrative Science Quarterly,* 38**,** 20-50.

HENDERSON-SELLERS, B. & SEROUR, M. K. 2005. Creating a dual-agility method: the value of method engineering. *Journal of Database Management,* 16**,** 1-23.

HUANG, J.-S., PAN, S. L. & LIU, J. 2017. Boundary permeability and online–offline hybrid organization: A case study of Suning, China. *Information & Management,* 54**,** 304-316.

HUGHEY, D. 2009. Comparing Traditional Systems Analysis and Design with Agile Methodologies. University of Missouri – St. Louis.

HUISMAN, M. & IIVARI, J. 2006. Deployment of Systems Development Methodologies: Perceptual Congruence Between IS Managers and Systems Developers. *Information & Management,* 43**,** 29-49.

IIVARI, J., HIRSCHHEIM, R. & KLEIN, H. K. 2000. A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches. *Journal of Management Information Systems,* 17**,** 179-218.

JANSEN, J. J., TEMPELAAR, M. P., VAN DEN BOSCH, F. A. J. & VOLBERDA, H. W. 2009. Structural Differentiation and Ambidexterity: The Mediating Role of Integration Mechanisms. *Organization Science,* 20**,** 797-811.

JOSHI, K. D., SARKER, S. & SARKER, S. 2007. Knowledge transfer within information systems development teams: Examining the role of knowledge source attributes. *Decision Support Systems,* 43**,** 322-335.

KANE, G. C. & ALAVI, M. 2007. Information technology and organizational learning: An investigation of exploration and exploitation processes. *Organization Science,* 18**,** 796-812.

KARLSSON, F. & AGERFALK, P. 2009. Exploring agile values in method configuration. *European Journal of Information Systems,* 18**,** 300-316.

KLEIN, H. K. & MYERS, M. D. 1999. A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly,* 23**,** 67-94.

KOTLARSKY, J. & OSHRI, I. 2005. Social ties, knowledge sharing and successful collaboration in globally distributed system development projects. *European Journal of Information Systems,* 14**,** 37-48.

LANGLEY, A. 1999. Strategies for Theorizing from Process Data. *Academy of Management Review,* 24**,** 691-710.

LEONARD, D. & SENSIPER, S. 1998. The role of tacit knowledge in group innovation. *California Management Review,* 40**,** 112-132.

LINDVALL, M., MUTHIG, D., DAGNINO, A., WALLIN, C., STUPPERICH, M., KIEFER, D., MAY, J. & KÄHKÖNEN, T. 2004. Agile Software Development in Large Organizations. *Computer,* 37.

LUBATKIN, M. H., SIMSEK, Z., LING, Y. & VEIGA, J. F. 2006. Ambidexterity and performance in small- to medium-sized firms: The pivotal role of top management team behavioral integration. *Journal of Management,* 32**,** 646-672.

MAHADEVAN, L., KETTINGER, W. J. & MESERVY, T. O. 2015. Running on Hybrid: Control Changes when Introducing an Agile Methodology in a Traditional "Waterfall" System Development Environment. *Communications of the Association for Information Systems,* 36**,** 77-103.

MARCH, J. G. 1991. Exploration and Exploitation in Organizational Learning. *Organization Science,* 2**,** 71-87.

MARUPING, L. M., ZHANG, X. & VENKATESH, V. 2009. Role of collective ownership and coding standards in coordinating expertise in software project teams. *European Journal of Information Systems,* 18**,** 355-371.

MCAVOY, J. & BUTLER, T. 2009. The role of project management in ineffective decision making within Agile software development projects. *European Journal of Information Systems,* 18**,** 372–383.

MCDERMOTT, R. & O'DELL, C. 2001. Overcoming cultural barriers to sharing knowledge. *Journal of Knowledge Management,* 5**,** 76-85.

MELNIK, G. & MAURER, F. 2004. Direct Verbal Communication as a Catalyst of Agile Knowledge Sharing. *Agile Development Conference.*

MILES, M. B. & HUBERMAN, A. M. 1994. *Qualitative Data Analysis,* Thousand Oaks, Sage Publications.

MOE, N. B., DINGSOYR, T. & DYBA, T. 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology,* 52**,** 480-491.

NERUR, S. & BALIJEPALLY, V. 2007. Theoretical Reflections on Agile Development Methodologies. *Communications of the ACM,* 50**,** 79-83.

NERUR, S., MAHAPATRA, R. & MANGALARAJ, G. 2005. Challenges of Migrating to Agile Methodologies. *Communications of the ACM,* 48**,** 73-78.

NEVIS, E. C., DIBELLA, A. J. & GOULD, J. M. (eds.) 2009. *Understanding Organizations as Learning Systems,* Wobum: Butterworth-Heinemann.

NEWELL, S. 2015. Managing knowledge and managing knowledge work: what we know and what the future holds. *Journal of Information Technology,* 30**,** 1-17.

NEWELL, S., HUANG, J. C., GALLIERS, R. D. & PAN, S. L. 2003. Implementing Enterprise Resource Planning And Knowledge Management Systems In Tandem: Fostering Efficiency And Innovation Complementarity. *Information and Organization* 13**,** 25-52.

NEWELL, S., ROBERTSON, M., SCARBROUGH, H. & SWAN, J. 2009. *Managing Knowledge Work and Innovation,* New York, Palgrave Macmillan.

NONAKA, I. 1994. A Dynamic Theory of Organizational Knowledge Creation. *Organization Science,* 5**,** 14-37.

NONAKA, I. & KONNO, N. 1998. The Concept of "Ba": Building a Foundation for Knowledge Creation. *California Management Review,* 40**,** 40-54.

O'REILLY, C. A. & TUSHMAN, M. L. 2008. Ambidexterity as a dynamic capability: Resolving the innovator's dilemma. *Research in Organizational Behavior,* 28**,** 185-206.

O'REILLY, C. A. & TUSHMAN, M. L. 2011. Organizational Ambidexterity in Action: How Managers Explore and Exploit. *California Management Review,* 53**,** 5-22.

OSHRI, I., VAN FENEMA, P. & KOTLARSKY, J. 2008. Knowledge transfer in globally distributed teams: the role of transactive memory. *Information Systems Journal,* 18**,** 593-616.

OZER, M. & VOGEL, D. 2015. Contextualized Relationship Between Knowledge Sharing and Performance in Software Development. *Journal of Management Information Systems,* 32**,** 134-161.

PERSSON, J. S., MATHIASSEN, L. & AAEN, I. 2012. Agile distributed software development: enacting control through media and context. *Information Systems Journal,* 22**,** 411-433.

PORT, D. & BUI, T. 2009. Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications. *European Journal of Information Systems,* 18**,** 317-331.

QUMER, A. & HENDERSON-SELLERS, B. 2008. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology,* 50**,** 280-295.

RAISCH, S., BIRKINSHAW, J., PROBST, G. & TUSHMAN, M. L. 2009. Organizational Ambidexterity: Balancing Exploitation and Exploration for Sustained Performance. *Organization Science,* 20**,** 685-695.

RAMESH, B., MOHAN, K. & CAO, L. 2012. Ambidexterity in Agile Distributed Development: An Empirical Investigation. *Information Systems Research,* 23**,** 323-339.

ROYCE, W. W. Managing the Development of Large Software Systems. Proceedings, IEEE WESCON, 1970. TRW, 1-9.

SAMBAMURTHY, V., BHARADWAJ, A. & GROVER, V. 2003. Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms. *MIS Quarterly,* 27**,** 237-263.

SANDBERG, J. & TSOUKAS, H. 2011. Grasping the logic of practice: theorizing through practical rationality. *Academy of Management Review,* 36**,** 338–360.

SCHATZKI, T. R., K., K.-C. & VON SAVIGNY, E. 2001. *The Practice Turn in Contemporary Theory* Taylor & Francis.

SCOTT, S. V. & WAGNER, E. L. 2003. Networks, negotiations, and new times: the implementation of enterprise resource planning into an academic administration. *Information and Organization,* 13**,** 285.

SERRADOR, P. & PINTO, J. K. 2015. Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management,* 33**,** 1040-1051.

STAKE, R. E. 2006. *Multiple Case Study Analysis,* New York, NY, The Guildford Press.

TURNER, N., SWART, J. & MAYLOR, H. 2013. Mechanisms for Managing Ambidexterity: A Review and Research Agenda. *International Journal of Management Reviews,* 15**,** 317-332.

TUSHMAN, M. L. & O'REILLY, C. A. 1996. Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change. *California Management Review,* 38**,** 8-30.

VERJANS, S. 2005. Bricolage as a way of life - improvisation and irony in information systems. *European Journal of Information Systems,* 14**,** 504-506.

VERSIONONE 2016. 10th Annual State of Agile Report.

VINEKAR, V., SLINKMAN, C. W. & NERUR, S. 2006. Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View. *Information Systems Management,* 23**,** 31-42.

WAGNER, E. & NEWELL, S. 2004. 'Best' For Whom?: The Tension Between 'Best Practice' ERP Packages And Diverse Epistemic Cultures In A University Context. *The Journal of Strategic Information Systems,* 13**,** 305-328.

WALSH, J. P. & SEWARD, J. K. 1990. On the Efficiency of Internal and External Corporate Control Mechanisms. *The Academy of Management Review,* 15**,** 421-458.

WANG, X., CONBOY, K. & CAWLEY, O. 2012a. "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *The Journal of Systems and Software,* 85**,** 1287-1299.

WANG, X., CONBOY, K. & PIKKARAINEN, M. 2012b. Assimilation of agile practices in use. *Information Systems Journal,* 22**,** 435-455.

WARD, K. 2004. Going global? Internationalization and diversification in the temporary staffing industry. *Journal of Economic Geography,* 4**,** 251-273.

WEST, D. & GRANT, T. 2010. Agile Development: Mainstream Adoption Has Changed Agility. *Forrester Research*.

WILLIAMS, L. & COCKBURN, A. 2003. Agile Software Development: It's about Feedback and Change. *Computer,* 36**,** 39-43.

YIN, R. K. 2009. *Case Study Research: Design and Methods,* Thousand Oaks, SAGE.

Appendix A – Coding Results

Knowledge Sharing Process	Interview Segments Coded	Document Segments Coded
Documentation	31	38
Requirements and Domain Knowledge	154	30
Training	81	11
Competence Management	50	27
Trust and Care	52	13
Team Composition	55	14
Continuous Learning	27	27
Knowledge Repositories	21	44
TOTAL	**471**	**204**

Appendix B – Knowledge Sharing Events in the First and Second Project Phases

Knowledge Sharing Process	First Project Phase	Second Project Phase
Documentation	A variety of documentation was in place to support the project, including detailed process workflows, project plans, and resource estimations. Although a service level agreement was established with the vendor, it was not viewed as being adhered to.	During the relaunch, formal documentation remained important. This included the creation of a project charter document, revised service level agreement, iteratively updated requirements document, and a technical strategy document signed off by the vendor.
Requirements and Domain Knowledge	An extensive evaluation of system requirements was conducted. Project team members demonstrated a thorough understanding of why the system was being implemented and what it needed to do. Although many requirements were met when the system went live, interviewees identified a variety of significant shortcomings.	The relaunch transitioned away from formal ownership of particular deliverables and increasingly relied on cross-functional discussions and decision making. This includes the adoption of a timebox approach, whereby requirements were prioritized and delivered in four week sprints.
Training	Highly structured, lecture-based, classroom training was used in the initial implementation. This approach was commonly viewed as valuable by the recruiters, but it was seen to take away from the day-to-day responsibilities of employees, who commonly work on commission. Training team turnover also negatively impacted the training quality.	The relaunch continued to employ interactive, hands-on training, supplemented by knowledge repositories. The training was derived from materials created for the first implementation, but about half was updated with the assistance of the IT department. The training was slightly more flexible, by allowing users more opportunities to experiment with the system features.
Competence Management	There was careful consideration taken during the vendor selection process, during data migration, and testing. However, the most significant issue was with the vendor, which was viewed by management as understaffed and unresponsive. Formal contracts and responsibilities were established with the vendor to control the deliverable quality.	The relaunch utilized more careful monitoring of project issues and more frequent meetings with the vendor. The client took the initiative to more thoroughly test code, independent of vendor testing. A technical strategy document was signed off by vendor and more rigorously defined quality criteria. Stand-up meetings improved the IT Project Manager's ability to address issues as they arose and find solutions.
Trust and Care	A hard deadline was established for the system go-live; however, interviewees disagreed on whether the system was ready to be implemented or if the date should have been pushed back. This appears to have created a level of distrust between executives and staff. Part of this issue stemmed from the lack of a clear business owner/champion for the project.	The relaunch demonstrated a more effective approach by the vendor to identify and resolve defects and deliver on needed functionality. Although the vendor was responsible for much of the coding, the company had extensive access to the source code to make data- and configuration-related changes to the system. Within the company's project team, tasks were

	As well, users and management perceived the vendor as not delivering the product that they had promised.	prioritized and the level of effort was estimated with poker cards and team members were given autonomy to sign up for their choice of tasks.
Team Composition	The project team was centralized and included representatives from across the organization. Subject matter experts participated as 'pilot team' members during requirements gathering and testing.	The relaunch restructured the project team into a more decentralized model. Team leaders were set up to oversee staffing deliverables, finance deliverables, and IT deliverables, with a ScrumMaster overseeing the project activities and backlog. This group ran as an agile team, but each team leader independently managed the activities related to their deliverable area.
Continuous Learning	Due to the initial performance and reporting issues with the system, users became increasingly familiar with the system functions through trial and error in the months following go-live.	Monthly sprint retrospectives were introduced for TechRecruit project team members. Discussions included what went well, issues that occurred, and opportunities for improvement. Team members became increasingly comfortable with agile techniques, such as planning poker.
Knowledge Repositories	Microsoft SharePoint was in place to store system related documents, but usage declined over the course of the project. An issue ticketing system was also in place, to keep track of ongoing bugs and resolutions related to the CRM.	The relaunch employed a more extensive, traceable repository of project knowledge thorough the use of the lightweight agile tool LeanKit. This included story cards and the project backlog. The tool was extensively used throughout the relaunch.

Appendix C – Supplementary Interviewee Quotes for the First and Second Project Phases

Knowledge Sharing Process	**First Implementation (Traditional)**	**Relaunched Implementation (Hybrid Agile-Traditional)**
Documentation	*'[The IT Project Manager] documented …the workflows, the dashboard, the set-up, the reports. We need more time to document the workflows of Finance and Accounting'.* -CIO	*'It would be email updates as well. So during that process that might take us three to four weeks during that to do all those things. We'd have weekly meetings and then after the meeting there'd be an email with what was discussed in that meeting and the requirements doc would be updated and then the next meeting we'd go in and discuss it and make sure.'* -VP Recruiting
Requirements and Domain Knowledge	*'I'm interested in data. How many clients you call on today? How many candidates you reach out? How many resumes have you read? Data are the output of their efforts. And I need their data to determine where we can be better and more efficient; where maybe we miss opportunities. I need data to value what we do better and what we don't.'* -Chief Marketing Officer	*'The essence [of agile development] is really the culture, the fit, getting people's head around that and you might find some people who although they are technically very talented, they're just unable to come to grips with working and being willing to share and be willing to extend themselves…the idea about agile is even though I'm a developer, I should be able to write functional tasks…if I'm a business development guy I should be able to do some UI work. It might be that I'm not a UI expert but now gaining some appreciation of what it takes to basically implement the user and integrate that with the business layer….So we've got that in terms of how to do that more effectively.'* -CIO

Training	*'The big challenge now is training people on [the new CRM system]. We have different people geographically located and also they have to take time to train getting away from their phone… they lose money! So what are you are going to do?...These guys are competitive, they are awesome. But they don't want to sit in a classroom and learn. The challenge is, how can we train? How to teach them? I need to be creative.'* -Training Specialist	*We put in exercises, like formal step one, try this, step two, do this, step three, do that. We'd have 10-15 steps or whatever on an exercise. But after I got done teaching I would tell them, 'listen, here's an exercise' and I would keep it showing on the projector but I would say, 'if you want to try something different because you want to see if it does this or does that you know, please feel free to go away from the procedure, you don't have to follow it step by step and letter by letter.'* -Training Consultant
Competence Management	*'The vendor for me was a big challenge. We had to manage a lot of people: our core team, internal IT Team, and the [vendor], who didn't have a Project Manager. The vendor is very small and they didn't have the resources that they originally planned. So externally the problem is to complete development with the vendor and keep on track of their deliverables. But, they are understaffed and you can't control it because they don't work for us.'* -IT Project Manager	*'We ended up buying a monitoring tool and implementing that just to watch and see and set up a baseline of certain things to happen within the guidelines of 0-2 seconds, 3-5 seconds and 5 seconds or longer. And anything that takes 5 seconds or longer would get flagged and we'd go and look at the processes that are running and such and share that information back and forth with [the vendor].* -VP Recruiting
Trust and Care	*'This project was bigger than we thought and there were not enough people working on it. People worked too many hours and they couldn't follow through on everything. Executives underestimated the impact that this change would have on the company. Why did Executives [decide to] go live?...The system was not ready to go-live for the back office, we needed more time and a better plan of attack.'* -Accounting Director	*Sometimes [the vendor] would agree to make a change and our IT Team does a great job and we have a test environment set up, they want to be able to test it to their liking before they're going to put it into production for us. And every once in a while we'd go to do something and it would lock up. And what happened was [that the vendor] would put something into production and we wouldn't know it. And you know, so there was no communication sometimes that way…There was a lot of times in meetings with them and on the phone and it was like 'You can't just do that to us, you've got to tell us…we want to be able to test it, but if we're not going to test it, you've at least got to tell us that you did it'.* -VP Recruiting
Team Composition	*'We have a cross-functional team. This is the way we collaborate, organize into groups, meet very regularly, and communicate very regularly. So there is collaboration between all the parts of the company.* -Chief Marketing Officer	*'So an example, I was the project manager for IT…so let's say one of the IT deliverables would be we had to create 15 laptops with a localized email environment and other configurations on the laptop specific for training purposes. So what I would then do is I would negotiate with my infrastructure team to get them to agree to a deliverable where I would actually set the expectations around requirements and what needed to get done. And then I would basically come back to [the Scrum Master] and say 'Okay, I'm committing for this sprint to get this effort done and it will be done within this four-week period of time'. So although the team who was acting on it wasn't really part of the Agile [team], I was part of the*

		Agile team and then I worked to extend myself and then to basically commit to doing what I needed to get done.' -CIO
Continuous Learning	*'[Business representatives have] meetings with the IT people every week. In these meetings, [the trainer] talks about issues and things that are updated, things that are changed in the system. We have to still work on the performance, workflows, re-calculator, search features.'* -Business Manager	*'We also did retrospectives about what we ran into in terms of issues, what we did well, what we didn't do well and why. There were issues we ran into and then we picked off one or two things that we thought we had to do better about.'* -CIO
Knowledge Repositories	*They've had a SharePoint system for a long time, it's a document management system but nobody uses it, nobody used it. And it's a very difficult system to find documents in because one of the interfaces for it is just to put in folders and files similar to Windows systems where they would just have to guess which path to go down, look in this folder then look in this folder then look in this folder and keep on following the path until they discovered, 'okay, it's not here'. Go down a different path to try to find something.* -Training Consultant	*'We looked into a couple software options to help streamline that process and…I came up with LeanKit because they had the whiteboard with the sticky notes, you could assign tasks to people, I could schedules certain things, we had pretty good analytics to see tasks being completed. What's reasonable for how many hours are allocated to each person and then I could track things that came up during our sprints that needed...I'd put them to a parking lot as they called it, because it needed to be vetted out further outside of our daily scrum.'* -IT Project Manager

Appendix D – Comparison of Research Contributions and Gaps

Authors/ journal	**Main goal of the paper**	**Main contribution of the paper**	**Remaining research gap and contribution by this research**
Ramesh et al., 2010, Organization Science	The study aims at shedding light on how organizations are able to incorporate both plan-driven and agile development methods to facilitate distribution and flexibility, respectively, which poses conflicting demands. In addition, the study aims to focus on how these conflicting demands can be addressed by 'balanced practices'.	Drawing on the ambidexterity literature, the authors identify specific practices, these including 1) operational focus; 2) formal structure but flexibility; 3) process assimilation before delivering quick value; 4) relational focus; 5) trust but verify; and 6) cohesive but distributed process teams. The overall contribution is theoretical and mainly addressed at extending the notion of ambidexterity in high level ISD practices.	The study focuses on high level practices while we dig deeper into specific knowledge sharing practices. While the authors assert that knowledge sharing is one of the key issues to consider in hybrid implementations, they do not provide specific details on how these are reshaped during a transition. We therefore build and extend this literature by identifying specific knowledge sharing practices. In addition, the authors call for research that takes their study further by focusing on whether ambidexterity in ISD projects leads to organizational performance. Our paper addresses this point as well.
Cao et al., 2009a, European Journal of Information Systems	The study aims at understanding how agile practices are appropriated by team members. Their goal is to apply adaptive structuration theory (AST)	A number of practices are identified that help 1) grounding process adaption and agile methods in theory; 2) extending AST; and 3) using AST in a novel	The study concentrates on the extent to which different contexts affect the way people learn practices. While the study has several merits it does not consider possible antecedents of learning agile (for instance, being

	to identify 'appropriated practices' that warrant successful absorption of agile approach, therefore they are looking at conditions where organizations are learning how to conduct agile ISD.	context. The underlying framework highlights that agile practices are adapted and appropriated based on the project, organizational, and development context.	already savvy in traditional approaches, as Ramesh et al. – and our paper – do). In addition, knowledge sharing aspects are mainly overlooked. Even if this does not represent a shortcoming of the paper (because of the different focus of this work), knowledge sharing aspects related to agile implementations are extremely relevant, as per what the literature suggest. Therefore here we build on this paper as we examine a particular context (people not agile savvy) and extend its findings by analyzing, in detail, knowledge sharing practices. In addition, they call for more theoretical focus when discussing agile-related ISD projects, and we believe to have taken this recommendation seriously.
Fitzgerald et al., 2006, European Journal of Information Systems	The study focuses on two particular agile methodologies: Scrum and XP. The author aim to test whether these two approaches are concurrent or complementary. The focus is on specific practices related to Scrum and XP and how they were implemented in an ongoing project (single study). the goals of the papers are 1) to investigate the usage and tailoring of agile methods in practice; and 2) to examine how agile methods could be combined to complement each other	The first contribution is descriptive and is about providing details on how the company they examine deviated, replaced and combined parts of Scrum and XP. The second contribution is more conceptual and relates to the identification of selected practices that need to be undertaken when attempting to combine these two agile methodologies.	The paper, similarly to ours, focuses on a single study and on specific practices that should be undertaken when combining two methodologies. However, first, Fitzgerald et al. concentrate on two agile methodologies we contrast traditional and agile approaches, at large. Second (and related to the previous point) while they focus on how management could help its teams learn how to combine Scrum and XP, we do so by examining how learning can be done, in ongoing processes, at the approach level (i.e., traditional vs. agile) rather than at the methodology level (i.e., Scrum vs. XP). Third, we focus on a particular aspect of agile ISD which is associated with knowledge sharing practices. Therefore, in this case too we complement and extend the literature.