HEC MONTRÉAL
École affiliée à l'Université de Montréal

Three essays on information systems development

par
Grégory Vial

Thèse présentée en vue de l'obtention du grade de Ph. D. en administration
(option Technologies de l'information)

Mai 2016

HEC MONTRÉAL
École affiliée à l'Université de Montréal

Cette thèse intitulée :

Three essays on information systems development

Présentée par :

Grégory Vial

a été évaluée par un jury composé des personnes suivantes :

Gilbert Babin
HEC Montréal
Président-rapporteur

Suzanne Rivard
HEC Montréal
Directrice de recherche

Alain Pinsonneault
McGill University
Membre du jury

Stacie Petter
Baylor University
Examinatrice externe

Jacques Robert
HEC Montréal
Représentant du directeur de HEC Montréal

Résumé

Les projets de développement de systèmes d'information (DSI) représentent une activité complexe et posant de nombreux défis aux entreprises. Afin d'améliorer l'efficience et l'efficacité de leur processus de DSI, ces dernières ont recours à différentes stratégies visant à optimiser l'organisation de ce processus. Cette thèse comporte trois essais, tous basés sur l'utilisation de la théorie organisationnelle comme discipline de référence afin d'améliorer notre compréhension, à travers l'utilisation de différentes perspective théoriques, de deux des plus importantes stratégies d'amélioration du processus de DSI : le DSI imparti et le DSI agile. Le premier essai propose l'élaboration d'une explication de type processus dans le contexte des projets de DSI impartis. Basée sur la théorie institutionnelle, cette explication permet de mieux comprendre les effets des différences entre parties impliquées dans ce contexte. Le deuxième essai porte sur le domaine du DSI agile et se base sur une étude de cas multiples afin d'étudier l'expérience que les équipes font de l'agilité dans le cadre de projets de DSI. À travers l'étude des tensions présentes dans ces projets, cet essai permet de mieux comprendre les défis auxquels font face ces équipes, les mécanismes pouvant contribuer à résoudre ces défis, ainsi que l'impact de ces mécanismes sur le projet. Le troisième essai, toujours sur le domaine du DSI agile, propose un développement conceptuel basé sur l'ontologie des routines organisationnelles développée par Feldman et Pentland. Cet essai propose une conceptualisation du processus de DSI comme routine organisationnelle et offre, dans le contexte du DSI agile, plusieurs avenues de recherche permettant de contribuer à l'amélioration de la compréhension de ce processus de DSI. Les trois essais composant cette thèse fournissent explications conceptuelles et empiriques rigoureuses sur des domaines d'actualité dans le domaine du DSI.

Mots clés : développement de système d'information, développement de système d'information imparti, théorie institutionnelle, explication processus, distance institutionnelle, demandes institutionnelles conflictuelles, développement de système d'information agile, routines organisationnelles, tensions, agilité, étude qualitative, étude de cas multiples.

Méthodes de recherche : étude qualitative, étude de cas multiples.

Abstract

Information systems development (ISD) projects are complex and challenging endeavors. Seeking to increase the efficiency and effectiveness of the ISD process, practitioners have relied on various strategies to organize ISD. This thesis is composed of three essays which all rely on organizational theory as a reference discipline to provide the theoretical lenses that can help us increase our understanding of two of those strategies: (1) outsourced ISD and (2) agile ISD. The first essay adopts a deductive approach and draws from institutional theory to build a process explanation on effects of differences between parties in outsourced ISD projects. The second essay focuses on agile ISD to investigate, using a multiple case study design, how project teams experience agility in ISD projects. The third essay adopts a deductive approach to propose a novel conceptualization of ISD as an organizational routine and explore the theoretical and empirical implications of this conceptualization using agile ISD as a contextual application. Overall the three essays that form this thesis address important questions to build theoretical as well as empirical insight in a rigorous manner on relevant topics of ISD research.

Keywords: information systems development, outsourced information systems development, institutional theory, process explanation, institutional distance, conflicting institutional demands, agile information systems development, organizational routines, tensions, agility, qualitative study, multiple case studies.

Research methods: qualitative study, multiple case studies.

Table of Contents

List of Tables

List of Figures

List of acronyms

ISD: Information systems development

OISD: Outsourced information systems development

Acknowledgments

Doctoral theses are officially presented as the results of the work of a single individual. The truth is, I find that my doctoral thesis is the result of a collaborative effort with a number of people who have supported me over the course of the past four and a half years. Beyond the actual product of this effort, it is the relationships with those people that I have come to value the most.

Professor Rivard, when I started my PhD, I knew it would be tough. But I did not realize it would be this tough. Without your help and support, this would have not happened. Thank you for everything you have done for me. You have been (and continue to be) an example of rigor, dedication and support that I can only hope to emulate. You have always been available and willing to help me even in times of tight deadlines, and in spite of all the other things you do. This has been a wonderful journey that has taught me so many things that I could probably write a thesis on it. “Tendu, mais tenable” will perhaps stick with me for a while…☺

I also would like to thank the members of my thesis committee. Professors Petter, Pinsonneault, Titah, and Babin, your insights on my work has been invaluable. I consider myself very lucky to have enjoyed discussions about my research (and other unrelated topics!) with you. Together as a committee, I think that you were what some software developers call a SWAT team, always available to provide help and support. This has extended beyond work where in the most casual conversations you have provided me with important pieces of advice that I will keep close to my heart for the rest of my career.

To my family, who have always supported me throughout this journey: Papa, maman, Thomas, Thibaut, et Benji, merci à vous tous! It has been hard and I have probably not kept in touch as much as I should have but you always remain close to my heart.

To Yuliya, my bright, funny and beautiful wife, as well as our two children, Timothée and Charlotte, who were born during the past four and a half years: thank you!. Baby, we both know that you are my better half and that without you, none of this would have happened. You have always supported and challenged me when I needed, forcing me to think and question my thoughts until not a doubt would remain. Thank you for everything, and thank you for our beautiful children. Timothée and Charlotte, pursue your dreams and remain true to yourselves, we love you very much and I thank you for always keeping me

grounded in the more important things in life. Zaya, you have seen me go through a Masters and a PhD, expecting nothing more than a pat (and a treat). Thank you for your affection and your loyalty.

I also would like to thank all the professors and staff at HEC Montreal, as well as those of the other universities in the joint PhD program. I have received a tremendous amount of support, encouragement, and help from professors who had no other interest in helping me than to see me succeed. This level of dedication has always amazed and continues to amaze me as I consider how valuable time is. Professors at HEC Montreal hold a special place in my heart as a group of wonderful, hardworking and extremely bright people.

A special thank you is due to my colleagues, with whom I have enjoyed countless conversations, whether about research, teaching, life or anything in between. Azadeh, Hanieh, Hoda, Jing Liu, Bogdan, Changan, Donald, Mahdi, Mahmood, Mathieu, Mohammad, Mustapha, Nasser, Reza, Thibaut as well as others with whom I have crossed paths: Thank you and I hope to have many more interesting conversations with you in the future.

And last, but definitely not least, I have been extremely lucky to have the attention, interest and support from a number of people with whom I have worked in the past or have had the chance to observe and interview for this thesis. I have learnt a great deal from all of you and your time has been extremely precious to me. I wish you all the best in all your professional and personal endeavors.

Preface

The three essays of this thesis are co-authored by Gregory Vial and Professor Suzanne Rivard.

Chapter 1: Overview of the Three Essays

1.1. Introduction

Information systems development (ISD) is one of the most significant sources of IT investments and concerns in organizations (Kappelman et al., 2014). As “the way in which information systems are conceived, analyzed, designed, and implemented” (Avison & Fitzgerald, 2006:23), ISD involves organizational actors working together and using various technologies to produce a software artifact that will eventually be deployed and used in the organization. Located at the junction between technological and social phenomena, ISD is a core phenomenon of interest for the IS discipline (Benbasat & Zmud, 2003).

In line with the concerns of practitioners, the literature on ISD has acknowledged the high levels of complexity (Avison & Fitzgerald, 2006) and uncertainty (Harris et al., 2009) associated with the ISD process. Typical challenges include rapid technological change (Lyytinen & Rose, 2006), unclear or changing requirements (Lee & Xia, 2010), difficulties working with vendors (Levina & Vaast, 2008) and transferring critical business knowledge across organizational boundaries (Dibbern et al., 2008). In addition, practitioners do not have the ability to rely on a “silver bullet” (Brooks, 1987) to structure work and ensure repeatable, successful outcomes across projects. As a result, many of those projects incur delays and costs overruns, system and information quality issues, implementation issues, as well as extensive legal proceedings in cases where vendors are involved. One recent high profile case highlighting those issues was the creation of the United States’ digital healthcare platform (Ford, 2013).

Aiming to addressing these issues, firms have turned toward approaches to increase the efficiency and effectiveness of the ISD process. Those approaches rely primarily on the implementation of changes in the way work is organized. In this thesis, we focus on two of those approaches.

The first approach is *outsourced ISD*, whereby a firm entrust the development of a system to an outside party. Outsourced ISD seeks to increase the efficiency and effectiveness of ISD by (1) enabling a client to gain access to technical or business expertise; and (2) using a vendor’s expertise to deliver a system at a lower cost.

Notwithstanding the popularity of the approach, several issues (e.g., communication, coordination) have been observed in outsourced projects, with differences between the parties involved deemed as important hinderers of outsourced ISD projects.

The second approach is *agile ISD*, which is based on the general concept of flexibility, defined as “the ability to improvise in reaction to changes” (Harris et al., 2009:401) applied to the context of ISD through agile ISD methods. Agile ISD methods are built on the Agile Manifesto (Beedle et al., 2001), an online document that promoted *agility* as an essential characteristic of successful ISD teams. While practitioners have embraced many features offered by those methods (VersionOne.com, 2013), research on the topic has highlighted a number of enduring issues (Abrahamsson et al., 2009; Conboy, 2009) that reflect some of the most pressing concerns expressed by practitioners on this topic (Agerfalk et al., 2009:322).

Aiming to increase our understanding of the implications of these two approaches, this thesis focuses on the study of: (1) outsourced ISD and (2) agile ISD, using organization theory as a reference discipline to provide the theoretical lenses to study them.

Essay 1 builds on the relevance of differences between the parties involved in an outsourced ISD project and noted in past empirical works on the topic to develop a *process explanation* (Poole et al., 2000) of the effects of those differences. Adopting institutional theory as a foundational theory and using a deductive approach, this essay unpacks the generic concept of *difference* to study the role of differing institutionalized practices as well as the logics that drive their enactment by parties. By focusing on the features of differences between parties in this context, Essay 1 provides important elements to explain the selection and enactment of various strategies parties may enact when differences between their respective practices become salient in the context of outsourced ISD.

Essay 2 draws from past research on agile ISD to address the question of “How do project teams experience agility in ISD projects?”. To answer this question a multiple case study design is used (Eisenhardt, 1989; Eisenhardt & Graebner, 2007). Drawing from the study of *tensions* in organization theory (Quinn & Cameron, 1988; Smith & Lewis, 2011), data analysis reveals that agility in ISD is experienced through the management of tensions (1) between principles of agility in ISD and the organization’s directives; and (2) within principles of agility in ISD. Acknowledging the false dichotomy between agility and the need to incorporate ISD projects within the broader context of the organization, Essay 2 will contribute to a more pragmatic understanding of the concept of agility in ISD by offering a theoretical explanation regarding the tensions faced by ISD teams, the mechanisms that contribute to address those tensions, as well as the impacts of those mechanisms on projects.

Essay 3 proposes to study the relationship between two opposite, yet complementary perspectives on ISD: one that embodies a generic, standardized business process and the other that conceptualizes ISD as a highly improvisational activity. Arguing that the complementary characteristic of these two conceptualizations is often overlooked in IS research, this essay proposes a novel conceptualization of ISD based on the ontology of organizational routines developed by Feldman and Pentland (2003). The theoretical insights brought by this perspective are discussed and its potential contribution to the empirical literature on ISD is evaluated through its application to the context of agile ISD. Building on five overarching issues on the topic, Essay 3 proposes six main avenues for research that have the potential to improve our understanding of this phenomenon while helping to address those issues. In the next sections, we provide a detailed presentation of each essay.

1.2. Summary of Essay 1

Title: “A Process Explanation of the Effects of Institutional Distance between Parties in Outsourced Information Systems Projects”.

Research Objective: Outsourced ISD (OISD) projects represent challenging endeavors because they are susceptible to issues encountered in ISD and outsourcing projects combined. These issues can impact the course of a project (Dibbern et al., 2008), its outcome (Natovich, 2003) as well as the relationship between the parties involved (Sabherwal, 1999). The empirical literature on the topic highlights the relevance of *differences between parties* as critical hinderers of OISD projects. The analysis of this literature reveals that three main gaps hinder our ability to build theoretical insight on the effects of those differences as well as the mechanisms that parties use to address them. Consequently, Essay 1 adopts a deductive approach to build an *explanation* (Gregor, 2006) of the effects of differences in OISD projects. To do so, institutional theory is used as a foundational theory to guide the development of four components that together form a *process explanation* (Poole et al., 2000) and provide a conceptual explanation of the strategies parties use to address differences between their respective institutionalized practices based on their nature as well as their saliency within the context of an OISD project.

Review of the Literature: Past works have highlighted a number of issues that can hinder the course of an OISD project. For instance the coordination of work in an outsourcing context may be hindered by cultural differences (Rai et al., 2009), or by work practice differences (Gowan & Mathieu, 2005) between the parties involved. In addition some of those issues are exacerbated by the geographical distance that exists between parties in an offshoring context. For example, poor infrastructure and time differences can hinder coordination and communication between parties (Bellah et al., 2013). A review of empirical works on OISD highlights a rich body of research grounded in a variety of theoretical lenses including control (e.g., Tiwana & Keil, 2009), coordination (e.g., Sabherwal, 2003), and culture (e.g., Winkler et al., 2008). The analysis of this literature reveals differences between parties involved in an OISD project as critical hinderers. Notwithstanding the advances made by the literature in improving our understanding of the OISD phenomenon, Essay 1 is motivated by the observation of three gaps. First, little is known on the mechanisms parties use to address differences throughout an OISD project, hindering our ability to understand how parties deal with differences in OISD projects. Second, there is a need to further unpack the notion of difference and account for the possibility that the logic underlying a difference between parties is not the same for both parties (e.g., culture for one party and laws for another), thereby helping us to better understand the incentives faced by parties to perform work in a certain manner. Third, differences between parties are often conceptualized and studied at a level (e.g., national culture) removed from that of the project even though they create issues in the project's daily activities.

Theoretical Development: Aiming at filling the gaps observed in the literature on OISD while answering recent calls for more theory development in IS (Markus & Saunders, 2007; Rowe, 2012), Essay 1 develops a *process explanation* (Poole et al., 2000) defined through the interaction of four components.

The first component is the construct of institutional profile as a characterization of each party involved in an OISD project and which acts as a series of initial conditions that parties bring into the project and carry throughout its course. This construct integrates dimensions of systems development (Avison & Fitzgerald, 2006) and project management (Project Management Institute, 2013) practices to inventory those practices institutionalized by each party along with their respective institutional *pillars* or logics of enactment.

The second component is the construct of institutional distance (Kostova, 1997) based on comparisons between the elements of parties' institutional profiles. These comparisons rely on the nature of the practices enacted by parties as they relate to a given dimension of their institutional profiles as well as the logics that drive the enactment of those practices (e.g., regulative versus normative).

The third component is the construct of conflicting institutional demands which emerge when the differences between parties' institutionalized practices become salient. While these differences may exist throughout the whole duration of a project, Essay 1 posits that some of those differences only become salient when parties realize that their respective institutional profiles are different and that the demands of the profiles are contradictory.

The fourth component is the repertoire of institutional strategic responses (Oliver, 1991) that can be enacted to address instances of conflicting institutional demands. Beyond the relevance of those strategies in the context of OISD projects, Essay 1 posits that a party's selection of a given strategy is explained by the constituting elements of the institutional distance between parties as well as that party's ability to envision collaboration with the other party beyond the project at hand.

Contributions: Essay 1 makes a theoretical contribution (Whetten, 1989) to the literature on OISD. Anchored in institutional theory, the explanation developed in Essay 1 adds to extant conceptualizations of differences between parties in OISD projects by explicitly theorizing on the properties of differences to better understand their effects. Being a process explanation, this work provides insight into the complex relationship between differences and difficulties in OISD projects, which have been largely ignored by past works, especially those relying on variance approaches. Accounting for the fact that the institutional logics that drive the enactment of institutionalized practices may differ between parties, Essay 1 offers a fine-grained explanation of the effects of differences between institutionalized practices in OISD projects based on their respective institutional pillars.

1.3. Summary of Essay 2

Title: “Experiencing Agility in Information Systems Development Projects”.

Research Objective: Recent research on ISD has highlighted the relevance of agile ISD methods as a means to structure work in a manner that allows actors to quickly react to changes (e.g., user requirements) impacting ISD projects. While there is a growing body of literature on this topic (Dingsøyr et al., 2012), there is still little agreement on the factors that influence the ability for ISD teams to be agile. Consequently, Essay 2 asks the research question: “how do project teams experience agility in ISD projects?”. Essay 2 builds on the large body of research on agile ISD method customization and tailoring (e.g., Fitzgerald et al., 2006) to argue that the difficulty to apply ready-made prescribed practices from those methods is the result of *tensions* and relies on a multiple case study design to empirically investigate them. The main contribution of Essay 2 is made to the literature on ISD by providing a *theoretical explanation* as to how ISD teams experience agility in projects through the study of tensions, mechanisms that contribute to address those tensions as well as the impacts of those mechanisms on ISD projects.

Review of the Literature: The literature on agility in ISD is largely based on the study of agile ISD methods. These methods find their roots in the Agile Manifesto, an online document dating back to 2001 and published by a group of software engineers (Beedle et al., 2001) advocating the need to incorporate *agility* in the ISD process. Today, it is widely accepted that agility in ISD is experienced through the enactment of practices prescribed by *agile ISD methods* founded on the Agile Manifesto (Conboy, 2009).

While agile methods such as eXtreme Programming (XP, Beck & Andres, 2004) and Scrum (Schwaber & Beedle, 2003) have enjoyed increasing popularity (VersionOne.com, 2013), this research is motivated by two main observations. First, there is an implicit assumption regarding the use of agile ISD methods and the experience of agility in ISD. We suggest that studying agility in ISD without focusing exclusively on methods may help us gain a better understanding of the challenges faced by teams in ISD projects as well the mechanisms that can help to address those challenges. Second, the literature has yet to explore the role of technical mechanisms to support agility in ISD. We suggest that accounting for mechanisms relying on technology in addition to those relying on interactions among actors, as is currently emphasized in the literature, may help us gain a better understanding as to how teams experience agility in ISD projects.

Research Methods: Given Essay 2's theory building objective, a multiple case study design following Eisenhardt's (1989) principles is used. To provide an "a priori specification of constructs" (p. 533) relevant to the research question, extant conceptualizations of agility in ISD are reviewed. In line with observations made in the IS literature and other fields, Essay 2 conceptualizes agility in ISD through its constituting facets: (1) Cooperation; (2) Flexibility; (3) Learning; and (4) Leanness.

Data collection was performed in five projects within three different organizations between January 2014 and September 2015. Following a review of ISD projects in each organization, access was granted to collect data for the cases. In four cases, the software was developed for internal customers while the fifth case represents the development of an artifact sold as software-as-a-service to external customers who are not involved in the ISD process. The types of IS developed are business intelligence systems and web-based transactional systems. Those systems were developed to support various functions in the target organizations such as marketing, logistics or accounting. Together those five cases present similarities and differences that ensure variation in the contexts where agility is sought by team members, revealing instances of tensions that are similar or different with one another.

Data analysis initially involved the use of grounded theory coding techniques consistent with the Glaserian approach (Glaser, 1978). More specifically, open coding was first used to allow for the maximum number of insight to emerge from the data in an unrestricted manner. Selective coding helped to categorize those initial coding instances against extant facets of agility in ISD and theoretical coding was used to create links between emergent concepts. This process will provided the data to perform within-case analysis of each of the five projects studied. Cross-case analysis was then performed to compare and contrast findings across cases (Eisenhardt, 1989).

Patterns uncovered from these analyses show that ISD teams face two main types of tensions: (1) between principles of agility in ISD and the organization's directives; and (2) within principles of agility in ISD. Those tensions, when unattended, have a variety of impacts on the project and the team, such as delays, extra work, or frustration. While social processes (e.g., educating business stakeholders) may help manage the first type of tensions, the analysis reveals that managing the second type is more complex because social processes may give rise to other tensions between principles of agility in ISD and the organization's directives. For instance, ensuring fast delivery and sound architectural principles may not be possible if actors in charge of the architecture are not committed to the project on a full-time basis. The analysis further shows that in some cases, technical processes (e.g., implementing continuous integration) play an instrumental role in addressing the second type of tensions.

Theoretical Development: Building on the patterns uncovered by the analysis of data, Essay 2 refers to the concept of *tensions* as "elements that seem logical individually but inconsistent and even absurd when juxtaposed" (Smith & Lewis, 2011:382). The theoretical arguments developed in Essay 2 are built on the evidence showing that experiencing agility in ISD is a complex process that extends beyond the simple application of practices prescribed by agile ISD methods. Rather, when trying to be agile in the course of a project, ISD teams must work within the context of the organization which may be at odds with some of the principles advocated by the Agile Manifesto and the practices prescribed by agile ISD methods. For instance, there may be specific guidelines with regards to software architecture at the organizational level that prevent the team from using a specific tool, programming framework or pattern to develop a software artifact quickly while following the principles of economy, simplicity and quality that drive leanness in agile ISD.

In addition, Essay 2 develops arguments with regards to the possibility that the principles of agility in ISD themselves are hard to reconcile because they may be inconsistent when taken together. For instance, while the Agile Manifesto advocates for "technical excellence" and the "best architectures", it also calls for the "early and continuous delivery of valuable software" as well as "welcom[ing] changing requirements".

Finally, Essay 2 theorizes on the interdependencies that may exist between tensions. While some tensions may be latent at the onset of a project, the mechanisms addressing other, salient tensions, may render those latent tensions salient. Essay 2 accounts for the possibility that addressing tensions among principles of agility in ISD using social mechanisms may in fact reveal other tensions with the rest of the organization.

Overall Essay 2 proposes that ISD teams experience agility in ISD through the management of tensions (1) between principles of agility in ISD and the organization's directives; and (2) within principles of agility in ISD.

Contributions: Essay 2 makes its main contribution to the literature on ISD. More specifically, Essay 2 proposes a theoretical explanation to help understand how project teams experience agility through the experience of various tensions throughout the course of an ISD project as well as the mechanisms that help to address those tensions. In addition, the a priori specification of the agility construct may serve as a basis for refining its conceptualization and operationalization.

1.4. Summary of Essay 3

Title: Conceptualizing Information Systems Development as an Organizational Routine: Implications and Avenues for Research

Research Objective: ISD projects represent complex and uncertain endeavors. Past works on ISD have typically approached this phenomenon from one of two main angles: (1) one where the ISD process is *structured* through ISD methods and frameworks; and (2) one where the ISD process is primarily emergent and improvisational. Acknowledging the insights yielded by these two perspectives, Essay 3 adopts a deductive approach and proposes to conceptualize ISD as an organizational routine based on the ontology proposed by Feldman and Pentland (2003) to account for the (1) stable, structured pattern of generic actions that make the ISD process as well as (2) the agency that actors exert as they perform the actions that form the ISD process. More specifically, Essay 3 posits that these two *aspects* are complementary rather than dichotomous and that their interaction allows for a process of structuration to occur. Building on these arguments, Essay 3 applies this conceptualization to the context agile ISD and proposes six main avenues for research on the topic that have the potential to increase our understanding of this phenomenon (Abrahamsson et al., 2009; Conboy, 2009).

Relevant Literature: While research has long acknowledged the complexity (Avison & Fitzgerald, 2006; Davis, 1982; Hirschheim & Klein, 1989; Kirsch, 1996) and uncertainty (Ang & Beath, 1993; Davis, 1982; Harris et al., 2009) associated with the ISD process, the literature on ISD has traditionally approached the phenomenon from one of two angles.

The *mechanistic perspective* conceptualizes ISD as a process made of a series of operations that are well-defined and carried out by actors who possess the business and technical expertise to fulfill them according to expectations (Brinkkemper, 1996). Conversely, the *emergent perspective* acknowledges the non-routine aspect of ISD and the need for actors to constantly improvise in their *practice* of ISD (Bansler & Havn, 2004) as they seek to *construct* a software artifact through the performance of socio-technical processes.

While the emergent perspective has enjoyed popularity in recent years, Essay 3 argues, in line with the mechanistic perspective, that there is a stable component to the ISD process and its enactment over time. For instance, ISD is often represented and communicated as a stable pattern of generic actions that together form the Software Development Life Cycle (SDLC). While methods may vary on their coverage of the various actions of the SDLC based on their focus, ISD always involves, at a high level, planning, analysis, design and implementation (Beck & Andres, 2004; Erickson et al., 2005) activities. In addition to this stable component, Essay 3 acknowledges the importance of the *performance* of ISD in line with the emergent perspective. Given the uncertainty and complexity of ISD, not all contingencies can be accounted for by artifacts such as ISD methods and frameworks that aim to structure the ISD process. In certain cases, the performances enacted as a result of those contingencies even carry the potential to alter the stable pattern of actions outlined by methods (e.g., Sarker & Sarker, 2009; Xu, 2009).

Theoretical Development: In line with the need to account for the stable component of ISD, its structuring power over performances as well as the emergent component of ISD that explains how performances depend on the ability for actors to improvise, Essay 3 develops a conceptualization of ISD based on Feldman and Pentland’s (2003) ontology of organizational routines. Organizational routines are defined as “repetitive, recognizable, patterns of interdependent actions, carried out by multiple actors,” (Feldman & Pentland, 2003:95) and are mutually constituted by two aspects. The *ostensive aspect* of a routine is based on a common, shared perception of the routine which may remain stable over time. The *performative* aspect of a routine represents the actual enactment of a given action for which actors may exert agency and choose from a “repertoire of possibilities” (Feldman & Pentland, 2003:102) while still being faithful to the ostensive aspect of the routine. In addition to these two aspects, artifacts represent an important component of organizational routines because they provide tangible references that help diffuse their ostensive aspect while enabling and constraining the performances of actors (Pentland & Feldman, 2005).

Applied to the context of ISD, Feldman and Pentland’s (2003) ontology accounts for the shared understanding that ISD is a process carried over time and involving a series of relatively generic actions (e.g., formulating requirements) while also accounting for the ability for actors to improvise when they perform ISD. The mutually constitutive aspects of organizational routines provide a theoretical foundation to explain how the ostensive aspect provides structure for the enactment of performance while performances in turn trigger the evolution of the ostensive aspect over time. The numerous software tools and documentation that actors rely on to guide and perform the ISD process provide artifacts that help characterize the ostensive and performative aspects of the ISD organizational routine.

To illustrate the potential of this conceptualization of ISD as an organizational routine, Essay 3 applies it to the context of agile ISD, a popular stream of research in ISD. Building on past research on the topic (Conboy, 2009; Dybå & Dingsøyr, 2008; Hummel, 2014), five enduring issues are identified based on their epistemological or phenomenological nature.

Contributions: Building on the work of Pentland and Feldman (2005), Essay 3 proposes six main avenues for research on the topic of agile ISD. The first avenue focuses on the study of performances and the influence of the context in their enactment as well as the implication of those performances on group and organizational level outcomes. The second avenue focuses on the ostensive aspect and the ability for multiple ostensive definitions of an organizational routine to exist across groups of actors and multiple levels. The third avenue focuses on artifacts and their role as illustrations of the ostensive and performative aspects of a routine. The fourth avenue focuses on the interaction between the ostensive and performative aspects of ISD their implication in the study of *transitions* between ISD methods and the overall evolution of the ISD organizational routine in organizations. The fifth avenue focuses on interactions between performances and artifacts and the enabling and constraining power that artifacts have on the enactment of patterns of actions by actors. The sixth avenue focuses on interactions between artifacts and the ostensive aspects of ISD and the implications of having multiple, concurrent ostensive aspects relying on the same artifacts to perform ISD and the impact of artifacts on those multiple ostensive aspects.

1.5. Conclusion and Contributions

Together the three essays that form this thesis propose a reflection on the richness of the ISD phenomenon using theoretical lenses borrowed from organization theory to build new insight on this topic. This thesis makes three main contributions to the literature on ISD: (1) an explanation on the effects of differences in OISD projects and the strategies parties may use to address them; (2) a theoretical explanation to help understand how project teams experience agility through the study of the tensions they face in the course of ISD projects; and (3) a conceptualization of ISD that proposes to stimulate research on ISD through the development of avenues for research in the context of agile ISD. Overall, the contributions of this thesis helps further our understanding of some important approaches used by practitioners to increase the efficiency and effectiveness of the ISD process.

References

ABRAHAMSSON P, CONBOY K and XIAOFENG W (2009) 'Lots done, more to do': The current state of agile systems development research. *European Journal of Information Systems, 18*, 281-284.

AGERFALK P, FITZGERALD B and SLAUGHTER S (2009) Flexible and distributed information systems development: State of the art and research challenges. *Information Systems Research, 20*(3), 317-328.

ANG S and BEATH CM (1993) Hierarchical elements in software contracts. *Journal of Organizational Computing and Electronic Commerce, 3*(3), 329-361.

AVISON D and FITZGERALD G (2006) *Information systems development: Methodologies, techniques & tools*. McGraw-Hill Education, New York, NY.

BANSLER J and HAVN E (2004) Improvisation in information systems development. In *Information systems research* (KAPLAN B, TRUEX D, III, WASTELL D, WOOD-HARPER AT and DEGROSS J, Eds), pp. 631-646, Springer US.

BECK K and ANDRES C (2004) *Extreme programming explained: Embrace change*. Addison-Wesley Professional, Upper Saddle River, NJ.

BEEDLE M, et al. (2001) Manifesto for agile software development. http://agilemanifesto.org/, accessed March 15, 2013.

BELLAH JC, BURNS JR and CASSIDY CM (2013) Offshore information systems development process in India: How practitioners respond to the challenges. *Journal of Information Technology Case and Application Research, 15*(2), 30-53.

BENBASAT I and ZMUD RW (2003) The identity crisis within the IS discipline: Defining and communicating the discipline's core properties. *MIS Quarterly*, 183-194.

BOEHM B and TURNER R (2004) Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *26th International Conference on Software Engineering (ICSE)*, pp. 718-719.

BRINKKEMPER S (1996) Method engineering: Engineering of information systems development methods and tools. *Information and software technology, 38*(4), 275-280.

BROOKS FP, JR. (1987) No silver bullet essence and accidents of software engineering. *Computer, 20*(4), 10-19.

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

CONBOY K and FITZGERALD B (2010) Method and developer characteristics for effective agile method tailoring: A study of xp expert opinion. *ACM Transactions on Software Engineering and Methodology, 20*(1), 2-30.

DAVIS GB (1982) Strategies for information requirements determination. *IBM Systems Journal, 21*(1), 4-30.

DIBBERN J, WINKLER J and HEINZL A (2008) Explaining variations in client extra costs between software projects offshored to India. *MIS Quarterly, 32*(2), 333-366.

DINGSØYR T, NERUR S, BALIJEPALLY V and MOE NB (2012) A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213-1221.

DYBÅ T and DINGSØYR T (2008) Empirical studies of agile software development: A systematic review. *Information and software technology, 50*(9/10), 833.

EISENHARDT KM (1989) Building theories from case study research. *Academy of Management Review, 14*(4), 532-550.

EISENHARDT KM and GRAEBNER ME (2007) Theory building from cases: Opportunities and challenges. *Academy of management journal, 50*(1), 25-32.

ERICKSON J, LYYTINEN K and SIAU K (2005) Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management, 16*(4), 88-100.

FELDMAN MS and PENTLAND BT (2003) Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly, 48*(1), 94-118.

FITZGERALD B, HARTNETT G and CONBOY K (2006) Customising agile methods to software practices at intel shannon. *European Journal of Information Systems, 15*(2), 200-213.

FORD P (2013) The obamacare website didn't have to fail. How to do better next time. http://www.businessweek.com/articles/2013-10-16/open-source-everything-the-moral-of-the-healthcare-dot-gov-debacle, accessed May 1, 2014.

GLASER BG (1978) *Theoretical sensitivity: Advances in the methodology of grounded theory*. Sociology Press, Mill Valley, CA.

GOWAN JA, JR. and MATHIEU RG (2005) The importance of management practices in IS project performance: An empirical study. *Journal of Enterprise Information Management, 18*(1/2), 235-255.

GREGOR S (2006) The nature of theory in information systems. *MIS Quarterly, 30*(3), 611-642.

HARRIS ML, COLLINS RW and HEVNER AR (2009) Control of flexible software development under uncertainty. *Information Systems Research, 20*(3), 400-419.

HIRSCHHEIM R and KLEIN HK (1989) Four paradigms of information systems development. *Communications of the ACM, 32*(10), 1199-1216.

HUMMEL M (2014) State-of-the-art: A systematic literature review on agile information systems development. In *38th Hawaii International Conference on System Sciences (HICSS)*, pp. 4712-4721, Big Island, HI.

KAPPELMAN L, MCLEAN E, JOHNSON V and GERHART N (2014) The 2014 sim IT key issues and trends study. [Article]. *MIS Quarterly Executive, 13*(4), 237-263.

KIRSCH LJ (1996) The management of complex tasks in organizations: Controlling the systems development process. [Article]. *Organization science, 7*(1), 1-21.

KOSTOVA T (1997) Country institutional profiles: Concept and measurement. *Academy of Management Proceedings, 1*, 180-184.

LEE G and XIA W (2010) Toward agile: An integrated analysis of quantitative and qualitative field data on software development quality. *MIS Quarterly, 34*(1), 87-114.

LEVINA N and VAAST E (2008) Innovating or doing as told? Status differences and overlapping boundaries in offshore collaboration. *MIS Quarterly, 32*(2), 307-332.

LYYTINEN K and ROSE GM (2006) Information system development agility as organizational learning. *European Journal of Information Systems, 15*(2), 183-199.

MARKUS ML and SAUNDERS C (2007) Editor's comments: Looking for a few good concepts...And theories...For the information systems field. *MIS Quarterly, 31*(1), iii-vi.

NATOVICH J (2003) Vendor related risks in IT development: A chronology of an outsourced project failure. *Technology Analysis & Strategic Management, 15*(4), 409-419.

OLIVER C (1991) Strategic responses to institutional processes. *Academy of Management Review, 16*(1), 145-179.

PENTLAND BT and FELDMAN MS (2005) Organizational routines as a unit of analysis. [Article]. *Industrial & Corporate Change, 14*(5), 793-815.

POOLE MS, VAN DE VEN AH, DOOLEY K and HOLMES ME (2000) *Organizational change and innovation processes: Theory and methods for research*. Oxford University Press, New York, NY.

PROJECT MANAGEMENT INSTITUTE (2013) *A guide to the project management body of knowledge: Pmbok® guide*. Project Management Institute.

QUINN RE and CAMERON KS (1988) *Paradox and transformation: Toward a theory of change in organization and management*. Ballinger, Cambridge, MA.

RAI A, MARUPING LM and VENKATESH V (2009) Offshore information systems project success: The role of social embeddedness and cultural characteristics. *MIS Quarterly, 33*(3), 617-A617.

ROWE F (2012) Toward a richer diversity of genres in information systems research: New categorization and guidelines. *European Journal of Information Systems, 21*(5), 469-478.

SABHERWAL R (1999) The role of trust in outsourced IS development projects. *Communications of the ACM, 42*(2), 80-86.

SABHERWAL R (2003) The evolution of coordination in outsourced software development projects: A comparison of client and vendor perspectives. *Information and Organization, 13*(3), 153-202.

SARKER S and SARKER S (2009) Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research, 20*(3), 440-461.

SCHWABER K and BEEDLE M (2003) *Agile software development with scrum*. Prentice Hall, Upper Saddle River, NJ.

SMITH WK and LEWIS MW (2011) Toward a theory of paradox: A dynamic equilibrium model of organizing. *Academy of Management Review, 36*(2), 381-403.

TIWANA A and KEIL M (2009) Control in internal and outsourced software projects. *Journal of Management Information Systems, 26*(3), 9-44.

VERSIONONE.COM (2013) Seventh annual state of agile development survey. http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf, accessed June 16, 2014.

WHETTEN DA (1989) What constitutes a theoretical contribution? *Academy of Management Review, 14*(4), 490-495.

WINKLER JK, DIBBERN J and HEINZL A (2008) The impact of cultural differences in offshore outsourcing - case study results from German-Indian application development projects. *Information Systems Frontiers, 10*(2), 243-258.

XU P (2009) Coordination in large agile projects. *The Review of Business Information Systems, 13*(4), 29-43.

XU P and RAMESH B (2007) Software process tailoring: An empirical investigation. *Journal of Management Information Systems, 24*(2), 293-328.

Chapter 2: Essay 1 – A Process Explanation of the Effects of Institutional Distance between Parties in Outsourced Information Systems Development Projects

Abstract

Outsourced information systems development (OISD) projects are challenging endeavors, and the literature suggests differences between the parties involved as critical hinderers of such projects. Using institutional theory as a foundational theory, we propose a process explanation of the effects of differences between parties in OISD projects. Our explanation relies on the interaction of four components: (1) the IS development and project management institutional profiles of the parties involved; (2) the institutional distance between practices within these profiles; (3) instances of conflicting institutional demands when institutional distance becomes salient, and (4) the repertoire of institutional strategic responses available to parties to address those instances. We suggest that the constitutive elements of institutional distance and the degree to which parties envision their collaboration beyond the project at hand contribute to explaining the enactment of strategic responses. Accounting for the fact that practices, as well as the institutional logics that drive their enactment, may differ between parties, we make a theoretical contribution to the literature on OISD by building a fine-grained explanation of the effects of differences between parties in OISD projects.

Keywords

Outsourced information systems development; institutional theory; process explanation; salience of institutional distance; conflicting institutional demands.

2.1. Introduction

Outsourced information systems development (OISD) projects are particularly challenging (Gopal & Koka, 2012), as they are susceptible to issues encountered in information systems development (ISD) and outsourcing endeavors combined. For instance, uncertainty in ISD projects renders collaboration toward a common goal difficult, as the goal itself is often ill-defined or changes with time (Beck & Andres, 2004). Similarly, the coordination of work in an outsourcing context may be hindered by cultural differences (Rai et al., 2009), or by work practice differences (Gowan & Mathieu, 2005) between the parties involved. It has been suggested that these combined issues can not only negatively impact the communication and trust between the parties involved in an OISD project, but can also threaten the outcome of the project itself (Verner & Abdullah, 2012) and eventually lead to long and expensive litigation (Natovich, 2003).

Our literature review on the challenges entailed by OISD projects highlights a rich body of research grounded in a variety of theoretical lenses including control (e.g., Tiwana & Keil, 2009), coordination (e.g., Sabherwal, 2003), and culture (e.g., Winkler et al., 2008). Our analysis of this literature reveals differences between parties involved in an OISD project as critical hinderers. Notwithstanding the advances made by the literature in improving our understanding of the OISD phenomenon, we observe three gaps motivating our work. First, little is known on the mechanisms (e.g., formal and relational control mechanisms) parties use to address differences throughout an OISD project. Second, there is a need to further unpack the notion of difference and account for the possibility that the logic underlying a difference between two parties is not the same for both (e.g., culture for one party and laws for another). Third, differences between parties are often conceptualized and studied at a level (e.g., national culture) removed from that of the project.

Aiming at filling these gaps and answering recent calls for more theory development in IS (Markus & Saunders, 2007; Rowe, 2012), we adopt institutional theory (Jepperson, 1991; Scott, 2008) as a foundational theory to conceptualize the effects of differences between parties in an OISD project. Ours is a process explanation (Poole et al., 2000) articulated by four components. The first component is the construct of institutional profile characterizing each party involved in an OISD project and which acts as a series of initial conditions that parties bring into the project and carry throughout its course. The second component is the construct of institutional distance between practices within parties' institutional profiles. The third component is the construct of conflicting institutional demands which emerge when the differences between parties' institutionalized practices become salient. The fourth component is the repertoire of institutional strategic responses (Oliver, 1991) that can be enacted to address instances of conflicting institutional demands.

Our work makes a *theoretical contribution* (Whetten, 1989) to the literature on OISD. Anchored in institutional theory, our model adds to extant conceptualizations of differences between parties in OISD projects by explicitly theorizing on the properties of differences to better understand their effects. Being a process explanation, our work provides insight into the complex relationship between differences and difficulties in OISD projects. Accounting for the fact that the institutional logics that drive the enactment of institutionalized practices may differ between parties, we offer a fine-grained explanation of the effects of differences between institutionalized practices in OISD projects, based on their respective institutional pillars.

The next sections review extant literature on OISD project hinderers, introduce our foundational theory and the four components forming our explanation. We then use these components to explain the enactment of strategic responses in an instance of conflicting institutional demands. We provide examples of the enactment of strategic responses to illustrate the versatility of our explanation before discussing the contributions of our work, its limitations, and its implications for future research and for practice.

2.2. Differences between Parties as OISD Project Hinderers

To identify studies pertaining to OISD project hinderers, we searched three databases (ProQuest, EBSCOHost, IEEE Xplore) from which we extracted peer-reviewed articles and conference proceedings, using combinations of keywords against abstracts. Keywords were selected to extract works on (1) IS projects involving (2) the development of an artifact (3) between a vendor and a client in an (4) outsourcing context. We carefully read the abstracts and the evidence provided to select sources pertaining to OISD projects and providing empirical evidence of difficulties encountered during the project. A process of backward and forward search (Webster & Watson, 2002) helped identify additional sources left out from initial search results to ensure a *reasonable* coverage of the literature on the topic (Rowe, 2014:246). This yielded 54 usable sources. We inventoried, for each source, the types of differences as well as the difficulties they caused, and organized this data in the form of a *concept matrix* (Webster & Watson, 2002) presented in Appendix 2.1.

Our review highlights a rich body of work that has used theoretical lenses such as transaction costs economics (e.g., Dibbern et al., 2008), control (e.g., Choudhury & Sabherwal, 2003), coordination (e.g., Sabherwal, 2003), and trust (e.g., Ali Babar et al., 2007). These works identify a number of difficulties related to, among others, communication (e.g., Bhat et al., 2006; Wareham et al., 2007), coordination (e.g., Sabherwal, 2003), or trust (e.g., Sabherwal, 1999), along with their antecedents, highlighting the complexity and uncertainty of OISD projects.

As synthesized in Table 2.1, our analysis reveals that *differences between the parties involved* emerge as key antecedents of OISD project difficulties. Although most of these works use the term *difference* (e.g., Bellah et al., 2013; Winkler et al., 2008), other terms such as *distance* or *barrier* (e.g., Dibbern et al., 2008) are also present, albeit less frequently. In other cases (e.g., Choudhury & Sabherwal, 2003; Zatolyuk & Allgood, 2004), the types of differences between parties emerge from the evidence provided even though authors may not explicitly name them.

Table 2.1. Differences between parties involved in an OISD project reported in the literature

Differences	Source(s)
Different cultures	
Different client-representative/project leader cultures	Rai et al. (2009)
Different national cultures	Ali Babar et al. (2007); Bellah et al. (2013); Bhat et al. (2006); Burmistrov (2006); Carmel and Agarwal (2001); Cusick and Prasad (2006); Davey and Allgood (2002); Dibbern et al. (2008); Dubé and Robey (1999); Gonzalez et al. (2010); Gregory et al. (2009); Herbsleb et al. (2005); Huang and Trauth (2007); Iacovou and Nakatsu (2008); Islam et al. (2009); Jain et al. (2011); Jarvenpaa and Mao (2008); Keil et al. (2007); Khan et al. (2011); Levina and Vaast (2008); Mahnke et al. (2008); Nakatsu and Iacovou (2009); Oza et al. (2004); Sabherwal (2003); Wareham et al. (2007); Winkler et al. (2008)
Different organizational cultures	Gregory et al. (2009); Herbsleb et al. (2005); Kannabiran and Sankaran (2011); Nicholson and Sahay (2001); Oza et al. (2004)
Different professional cultures	Herbsleb et al. (2005); Huang and Trauth (2007)
Different work cultures	Bhat et al. (2006); Dubé and Robey (1999)
Different currencies	Nakatsu and Iacovou (2009)
Different geopolitical environments	Gonzalez et al. (2010); Nakatsu and Iacovou (2009); Natovich (2003)
Different interests	Abdullah and Verner (2012); Bhat et al. (2006); Gopal and Koka (2010); Mirani (2007); Natovich (2003); Sabherwal (1999); Sabherwal (2003)
Different languages	Ali Babar et al. (2007); Arnott et al. (2007); Bellah et al. (2013); Bhat et al. (2006); Davey and Allgood (2002); Gonzalez et al. (2010); Huang and Trauth (2007); Iacovou and Nakatsu (2008); Islam et al. (2009); Khan et al. (2011); Levina and Vaast (2008); Nakatsu and Iacovou (2009); Oza et al. (2004); Pries-Heje et al. (2005); Sabherwal (2003)
Different legal environments	Bellah et al. (2013); Frank (2005); Jarvenpaa and Mao (2008); Khan et al. (2011); Mathew (2011); Nakatsu and Iacovou (2009); Pries-Heje et al. (2005)
Different locations	Abdullah and Verner (2012); Bellah et al. (2013); Bhat et al. (2006); Burmistrov (2006); Cusick and Prasad (2006); Dibbern et al. (2008); Huang and Trauth (2007); Iacovou and Nakatsu (2008); Nicholson and Sahay (2001)
Different maturity levels	Wareham et al. (2007)
Different methodologies	Abdullah and Verner (2012); Bhat et al. (2006)

Different organizations	Ahuja and Sinclair (2012); Bellah et al. (2013); Benito et al. (2013); Carmel and Agarwal (2001); Choudhury and Sabherwal (2003); Cusick and Prasad (2006); Dibbern et al. (2008); Gopal et al. (2002); Gopal and Gosain (2010); Gopal and Koka (2012); Gopal et al. (2011); Gregory et al. (2009); Gregory et al. (2013); Herbsleb et al. (2005); Iacovou and Nakatsu (2008); Islam et al. (2009); Jain et al. (2011); Kannabiran and Sankaran (2011); Khan et al. (2011); Levina and Vaast (2008); Mirani (2007); Nakatsu and Iacovou (2009); Nurmi et al. (2005); Oza et al. (2004); Palvia (2008); Roy and Aubert (2002); Sabherwal (2003); Tiwana (2004); Tiwana (2010); Verner and Abdullah (2012)
Different technological infrastructures	Bellah et al. (2013); Gonzalez et al. (2010); Islam et al. (2009); Nakatsu and Iacovou (2009); Pries-Heje et al. (2005); Zatolyuk and Allgood (2004)
Different technologies	Nurmi et al. (2005)
Different time zones	Bellah et al. (2013); Carmel and Agarwal (2001); Cusick and Prasad (2006); Gonzalez et al. (2010); Gopal et al. (2011); Herbsleb et al. (2005); Kannabiran and Sankaran (2011); Nakatsu and Iacovou (2009); Nicholson and Sahay (2001)
Different tools	Bhat et al. (2006)
Different work practices	Benito et al. (2013); Gowan and Mathieu (2005); Gregory et al. (2013); Gregory et al. (2009); Mahnke et al. (2008); Nakatsu and Iacovou (2009); Natovich (2003); Rai et al. (2009)

In the majority of the studies we reviewed (36 out of 54) parties face more than one type of difference hindering their collaborative effort although the emphasis may be set on one specific type of difference. For instance, OISD projects in offshore settings often involve a combination of differences in national cultures and technological infrastructure (e.g., Gonzalez et al., 2010; Nakatsu & Iacovou, 2009) that hinder communication between parties.

While some differences result from the physical distance between parties (e.g., differences in geopolitical environments), in most cases – even in those involving physical distance – organizational distance (e.g., differences in organizational cultures) appears as a key antecedent of OISD project difficulties. For example, many works (e.g., Dibbern et al., 2008; Levina & Vaast, 2008) highlight the difficulties associated with the successful transfer of important domain knowledge across organization boundaries, regardless of the physical location of the other party.

Notwithstanding these insights, we identified three main gaps in the literature, echoing some of the conclusions made by recent works on the topic (e.g., Beck, 2014; Gregory et al., 2013). First, little is known on the mechanisms parties use to address differences during a project. Addressing the concerns of practitioners on the topic, the literature emphasizes the prediction and prescription of mechanisms to alleviate issues in OISD projects. More specifically, two main categories of *control* mechanisms are studied: (1) formal control mechanisms (e.g., legal contracts) and (2) relational mechanisms which rely on the expectations and obligations parties form toward one another (e.g., Jain et al., 2011; Koh et al., 2004). For example, Huber et al. (2013) studied four OISD projects undertaken by a German bank to build a process model uncovering the substitution and complementarity of formal and relational governance mechanisms based on different patterns of events. Notwithstanding these contributions, we are still missing insight into the contextual links between differences between parties and the various mechanisms parties use to try to address those differences at different points in time.

Second, there is a need to further unpack the notion of *difference* and account for the possibility that the logic underlying a difference between parties is not the same for both parties (e.g., culture for one party and laws for another). Indeed, qualifying a difference as “cultural” assumes that culture is at the root of a difference between a client and a vendor. However, a client may work in a particular manner for cultural reasons while a vendor may do so due to their legal environment. For example, a client may enforce eXtreme Programming's forty hour work week (Beck & Andres, 2004:41) because they have formally adopted that method while a vendor may only do so because of the labor laws of the country where they operate. Toward that end, some works rely on the individual components of high-level constructs to propose a more refined explanation of the effects of differences between parties. For instance, Gregory et al. (2009) have studied the role of the components of cultural intelligence – cognitive, motivational and behavioral – in the emergence of a negotiated culture between parties. In line with this work and others that rely on multiple theoretical lenses to account for the complex nature of differences between parties in OISD projects (Dibbern et al., 2008; Gonzalez et al., 2010), we argue that accounting for the respective logics underlying differences in the context of an OISD project may help better understand the effects of those differences as parties may not face the same incentives to perform work in a certain manner.

Finally, our analysis of the literature highlights a need to better understand how differences between parties conceptualized at a high level (e.g., national culture) can impact what Beck (2014) refers as "daily project business" (p. 245). Consistent with the objective of generalization sought by predictive and prescriptive models, differences may be conceptualized as a generic lack of congruence for a given element characterizing the parties involved. This is the case of studies that use aggregate constructs such as national culture where the links between those high-level constructs and the execution of work is not always clear. For example, Rai et al. (2009) studied cultural differences between parties based on the five national cultural values measured in Hofstede et al. (1990) to predict project success. While their results show that uncertainty avoidance plays an important role in a project's cost overruns and client satisfaction, it is unclear how each party's level of uncertainty avoidance becomes salient and results in differences in the way they perform work. Indeed, in the case of national culture – one of the most frequently reported difference in OISD projects as illustrated in Appendix 2.1 – this issue has been raised in works outside of the IS community (Shenkar et al., 2008) as well as within the OISD literature (Beck, 2014). In line with this gap, Beck (2014) finds that "conflict in global ISD outsourcing projects nowadays relates more to issues involving the daily project business rather than cultural differences" (p. 245). We argue that studying differences between parties in an OISD project requires the use of theoretical lenses which remain rooted in the day-to-day execution of that project. However, we also extend Beck's (2014) arguments and posit that theoretical lenses that account for the effects of differences between parties *across levels* (e.g., how different legal environments translate into differing practices) have the potential to yield important insight on this phenomenon.

Building on the relevance of differences between parties as the main hinderers in OISD projects, we anchor our theory building effort in institutional theory. In doing so, we focus on the role of institutionalized practices in OISD projects and study – under a common set of foundational assumptions – their role as sources of differences between parties. We then build on this foundation to explain the effects of institutional distance during the course of an OISD project.

2.3. Institutional Theory as a Foundational Theory

We adopt institutional theory (e.g., Jepperson, 1991; Scott, 2008) as a foundational theory because it offers important concepts to characterize the differences highlighted in the literature on OISD projects and explain their effects. An institution represents "a social order or pattern that attained a certain state of property" (Jepperson, 1991:145). Institutions are stable entities. Although they may become deinstitutionalized at the expense of new institutions, this process occurs over long periods of time (e.g., Greenwood et al., 2002). Institutions materialize under different forms, such as practices, social norms, and rules. We focus on institutions as social structures made of patterns of legitimate behaviors that both constrain and enable actors through the exertion of pressure to foster compliance (Jepperson, 1991; Scott, 2008). The notion of continued enactment of prescribed behaviors relies on the assumption that actors seek legitimacy and will comply with the pressure (Scott, 2008). Legitimacy is defined as a "generalized perception or assumption that the actions of an entity are desirable, proper, or appropriate within some socially constructed system of norms, values, beliefs and definitions" (Suchman, 1995:574).

An institution is a relative, rather than an absolute concept, with relativity comprising four dimensions (Jepperson, 1991:146). To illustrate each dimension, we use the example of North American public accounting firms, which are oft cited as highly institutionalized entities (e.g., Carpenter & Feroz, 2001). The first dimension pertains to the context: "whether a practice is an institution is relative to particular contexts." For example, the International Financial Reporting Standards (IFRS) are an institution for accounting firms in the European Union and Canada, but not in the United States where they have not been adopted as a standard. The second dimension of relativity pertains to levels within a given context: "primary levels of organization can operate as institutions relative to secondary levels of organization." For example, the audit procedures developed by the audit department of an accounting firm are an institution for all the audit teams across the organization. The third dimension refers to the notion that "whether an object is an institution is relative to a particular dimension of a relationship." In a Canadian public accounting firm, the Canadian Generally Accepted Auditing Standards (GAAS) represent more an institution for employees of its audit department than for employees of its tax department, who consider Canadian tax laws more an institution than GAAS. The fourth dimension is that "whether an object is an institution is relative to centrality." For instance, tax laws represent rules that a majority of tax payers follow without questioning while tax experts often question their applicability. For researchers, the relativity of institutions implies the identification of the context within which a structure, a practice or an object is considered an institution.

It has been suggested that the logic of enactment of institutional behaviors, the *pillar* (Scott, 2008) that appeals to the actor's sense of obligation, works in one of three different ways. Under the *regulative* pillar, behavior is enacted based on established rules associated with rewards and sanctions that coerce actors into compliance. In the context of a country, regulative institutions are based on laws and regulations. For organizations, legal contracts are based on a regulative pillar as they define rules and associated sanctions (e.g., late delivery penalties) as well as rewards (e.g., a payment schedule). A *normative* pillar refers to the norms prevailing in a particular context. For example, as one of the knowledge areas of the Project Management Institute (PMI), risk management is a norm for project managers who are certified by the PMI. Under a *cultural-cognitive* pillar, institutions are enacted based on the existence of a common frame through which actors make sense of the world. For example, employees in many North American organizations follow "casual Friday" where they dress less formally on Fridays than during the rest of the week.

In IS, institutional theory has been found useful to explain a variety of phenomena (Mignerat & Rivard, 2009), including the institutionalization process of IS project management practices (Mignerat & Rivard, 2012), the role of institutional pressures to explain technology adoption (e.g., Teo et al., 2003), outsourcing decisions (Ang &

Cummings, 1997), or the possible misalignments between an IT and the institutions in place within an organization (e.g., Barca & Cordella, 2006).

In OISD, we argue that several types of differences observed in past empirical works are embedded within differing institutionalized practices enacted by parties. For instance, cultural values influence the way an organization's employees communicate and coordinate their work (e.g., Levina & Vaast, 2008; Walsham, 2002). From an institutional perspective, the enactment of those practices is explained by their relevance as cultural-cognitive institutions in the organization. Similarly, the adoption of an ISD method shapes ISD practices and over time, defines a normal way of developing software for an organization (Fitzgerald et al., 2002). From an institutional perspective those practices that have become norms in the organization are embedded in normative institutions. Finally, the rules and regulations of the environment within which an organization operates influences the practices its employees must enact to avoid sanctions. Under an institutional explanation, these practices are studied as regulative institutions. We argue that an institutional perspective on OISD projects allows us to address the gaps we have identified in the literature. The first gap – little knowledge on the mechanisms parties use to address differences during a project – is addressed by the array of strategies institutional actors may use in response to institutional pressures (Oliver, 1991). The second gap – the need to further unpack the notion of difference and account for the possibility that the logic underlying a difference between parties is not the same for both parties (e.g., culture for one party and laws for another). – is addressed by the ability for institutional theory to account for the institutional logic driving the enactment of practices by a given party. The third gap – the conceptualization and study of differences at a level often removed from that of the project – is addressed by the importance of management practices as institutions in organizations (Jepperson, 1991) and their relevance in the context of ISD projects (Mignerat & Rivard, 2012).

2.4. Theoretical Development

We start by outlining the boundary conditions under which our explanation holds. First, consistent with the role of differences as critical hinderers in OISD projects, the enactment of practices as a focal point of research on ISD (Avison & Fitzgerald, 2006) and the conceptualization of institutions as patterns of prescribed behaviors enacted by institutional actors (Jepperson, 1991; Scott, 2008), we focus on the role of differences between institutionalized practices in OISD projects. Institutionalized practices are defined as "particular ways of conducting organizational functions that have evolved over time under the influence of an organization's history, people, interests and actions that have become institutionalized in the organization" (Kostova, 1999:309). In the IS literature, a set of practices is said to be institutionalized when their legitimacy is recognized in organizations where they have "acquired the status of norms or quasi-rules" (Mignerat & Rivard, 2012:126). Second, in line with the theoretical (e.g., Scott, 2008) and empirical (e.g., Avgerou, 2000; Greenwood et al., 2002) arguments toward the stability of institutions, we assume that institutions will remain stable for the duration of a project because projects are not likely to last long enough to allow for deinstitutionalization and reinstitutionalization. This important assumption of institutional theory differs from that of other perspectives, such as the practice perspective, wherein practices are highly situated and emerge over time from the actions of the people that engage in the practice (see Feldman & Orlikowski, 2011). In this perspective, practices change often simply because they are performed (Feldman & Pentland, 2003). Third, for the sake of parsimony we limit our explanation to projects that involve only two parties, a client and a vendor. Fourth, our explanation pertains to ISD projects where a software artifact is built, configured or maintained. Finally, we assume that the project requires regular exchanges between parties over its course, thus excluding projects characterized by an arm's length relationship between client and vendor.

Our primary goal is to propose an *explanation* of the effects of differences between parties in an OISD project, that is, to "promote greater understanding or insights by others into the phenomenon of interest" (Gregor, 2006:619). To do so, we build a "process explanation" (Poole et al., 2000:17) of how parties react to differences with other parties in an OISD project. Process explanations rely on a definition of causality wherein a *combination* of elements provides the theoretical links to explain a phenomenon (Poole et al., 2000). We argue – along with the literature on OISD – that differences between parties are critical hinderers of OISD projects. Those differences are especially relevant in this context because ISD projects are complex and uncertain and often require closer collaboration between parties than other types of outsourcing projects (e.g., infrastructure). We treat these differences as events which trigger a process aimed at addressing those differences through a variety of response mechanisms enacted by parties. More specifically, our explanation, illustrated in Figure 1, is based on the interaction of four components. These components are used to (1) characterize the institutionalized practices of the parties involved; (2) compare those practices across parties; (3) understand when differences between institutionalized practices become problematic through their salience; and (4) explain how parties respond to those problematic scenarios based on the nature of those differences.

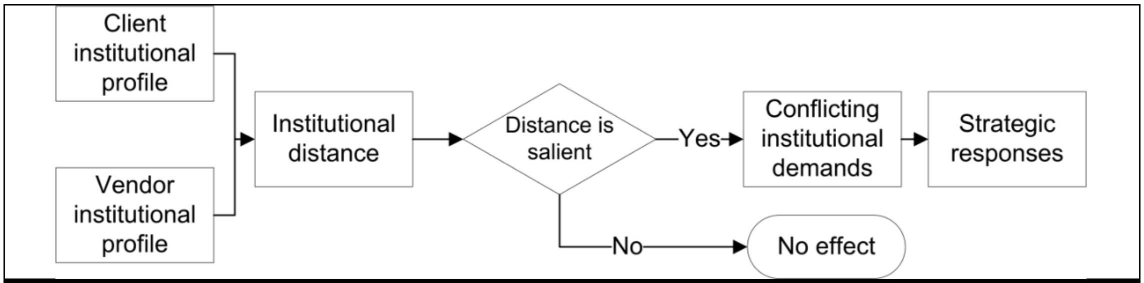

Client institutional profile
Vendor institutional profile
Institutional distance
Distance is salient
Yes
Conflicting institutional demands
Strategic responses
No
No effect

Figure 2.1. A process explanation of the enactment of responses to conflicting institutional demands in OISD projects

The concept of salience, which is an explicit part of our explanation, accounts for the possibility that not all differences between parties may be relevant throughout the course of a project or at specific points in time (e.g., Gregory et al., 2013). Indeed, the literature on OISD has traditionally conceptualized differences as sources of quasi permanent issues (e.g., Rai et al., 2009). For instance, our analysis of the literature shows the importance of differences in national cultures as sources of difficulties between parties. However, not all differences between national cultures may be relevant at any given point in time in an OISD project. More importantly, it has been suggested that despite significant differences in national cultures between two parties, the emergence of standardized training procedures may diminish the impact of those differences. In his study of a large IT offshoring project between a German bank and an Indian vendor, Beck (2014) concludes that "the worldwide alignment of professional practices, and a maturing market for ISD offshoring services increasingly are leading to fewer culture-related conflicts in global IS sourcing projects" (p. 245). In line with this argument, we suggest that the relevance of differences between parties in an OISD project should be primarily based on their salience and their impact on the "daily project business" (Beck, 2014:245) rather than their mere presence. We also posit that during the course of a project, there are multiple occasions where differences become salient and for which our explanation applies while attempts at addressing issues caused by those differences may not always lead to their resolution (e.g., Natovich, 2003).

In the next sections, we present the four components that form our explanation. Together these components explain how, why and when parties respond to differences between their institutionalized practices. Throughout the development of our explanation , we follow recommendations on theory building by providing illustrative examples (Jaccard & Jacoby, 2009; Rivard, 2014).

2.4.1. Component 1: Institutional Profiles as Characterizations of the Parties' Institutionalized IS Development and Project Management Practices

An ISD project involves two main categories of practices: systems development and project management practices (Avison & Fitzgerald, 2006). To characterize a set of relevant IS development practices, we adopt Avison and Fitzgerald's framework (2006:598) which proposes seven elements to compare system development methodologies, techniques and tools (see Table 2.2). The elements that compose this framework have been useful in past works to compare ISD methodologies (e.g., Torkar et al., 2011). In terms of project management practices, we rely on the ten knowledge areas identified by the PMI (2013) – listed in Table 2.2 – which were found relevant to characterize IS project management practices (Mignerat & Rivard, 2012).

Table 2.2. ISD related practices

Elements of IS Development Methodologies, Techniques and Tools
(adapted from Avison & Fitzgerald, 2006:598-603)

Philosophy:
The foundation upon which a problem is approached and a solution is crafted, e.g., fulfilling customer's requirement or developing a system on time and on budget.

Model:
The methods of representation of the solution; e.g., process or class diagrams

Techniques and Tools:
The technical choices in terms of systems development; e.g., rapid application development (RAD) tools and generic software patterns libraries.

Scope:
The activities involved in each phase of the systems development; e.g., the scope of these activities in a waterfall project from an Agile project.

Outputs:
The deliverables produced at the end each stage, including the final deliverable; e.g., functional requirements produced during the early stages of the project.

Practice [of ISD]:
How ISD activities are experienced by stakeholders; e.g., difficulties encountered in applying standard activities in unknown contexts and the ability to modify them accordingly.

Product:
Artifacts obtained through the adoption of the methodology and that support its use; e.g., documentation manuals, training, software.

Project Management Knowledge Areas
(Project Management Institute, 2013)

Integration:
"Processes and activities needed to identify, define, combine, unify, and coordinate the various processes and project management activities" (p. 554).

Scope:
"Processes required to ensure that the project includes all the work required, and only the work required" (p. 555).

Time:
"Processes required to manage the timely completion of the project" (p. 556).

Cost:
"Processes involved in planning, estimating, budgeting, financing, funding, managing, and controlling costs" (p. 553).

Quality:
"Processes and activities of the performing organization that determine quality policies, objectives, and responsibilities" (p. 555).

Human Resources:
"Processes that organize, manage and lead the team" (p. 554).

Communication:
"Processes that are required to ensure timely and appropriate planning, collection, creation, distribution, storage, retrieval, management, control, monitoring, and the ultimate disposition of the project information" (p. 553).

Risk :
"Processes concerned with conducting risk management planning, identification, analysis, response planning, and controlling risk on a project" (p. 555).

Procurement :
"Processes necessary to purchase or acquire the products, services, or results needed from outside the project team" (p. 555).

Stakeholder:
"Processes required to identify all people or organizations impacted by the project, analyzing stakeholder expectations and impact on the project, and developing appropriate management strategies for effectively engaging stakeholders in project decisions and execution" (p. 556).

We propose that each party of an OISD project is characterized by an *institutional profile* made of institutionalized practices – as represented in Figure 2.2 – each being associated with regulative, normative, or cultural-cognitive pillars. For example, an organization may mandate the use of UML sequence diagrams through the use of standards and specific software customized with specific templates that prevent the use of other languages. In this case, institutionalized practices that relate to the *model* dimension will inventory the use of UML sequence diagrams while the *techniques and tools* dimension will inventory the use of a customized software to create mandated diagrams. In both cases, the practices rely on a normative pillar.

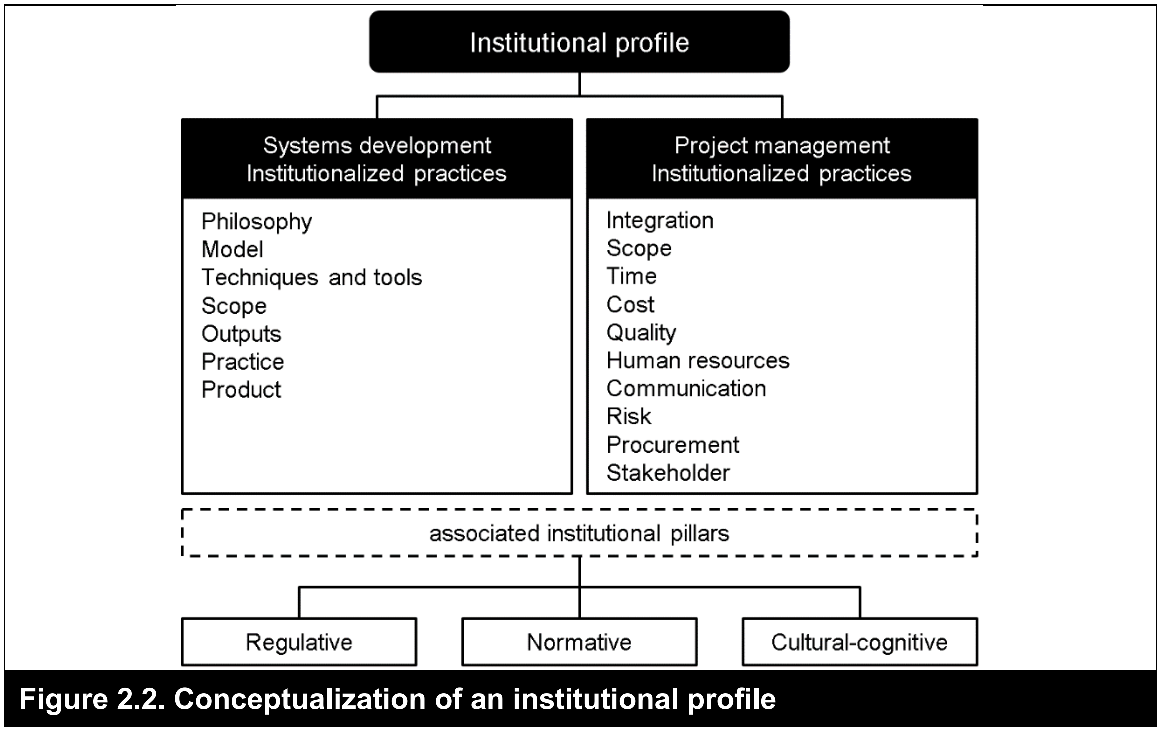


Figure 2.2. Conceptualization of an institutional profile

The construct of institutional profile has been introduced in the context of the transfer of strategic practices within multinationals (e.g., Kostova, 1997, 1999). Although this literature does not explicitly define the construct, it uses the notion of country institutional profile to refer to institutions in place within each organization's country as they relate to the three institutional pillars. In IS research, one study refers to the concept of institutional profile, yet without structuring it around a set of dimensions (Ayres, 2003). Instead, it refers to two general institutional profiles, a "process maturity" profile based on the CMM and a "responsiveness" profile aimed at accommodating tight deadlines and changing requirements.

Because of the institutional actors' search for legitimacy, the presence of institutionalized IS development and project management practices suggests that a number of the ISD actors' behaviors will reflect their compliance with the institutions in place within their organization. This is illustrated by the president of a software vendor commenting on how ISD quality standard practices are enacted based on a cultural-cognitive foundation: "We do not draw the picture of high quality standards but we follow them" (Vial, 2009:92).

2.4.2. Component 2: Institutional Distance as Manifestation of Differences between Practices within Institutional Profiles

In an OISD project, the parties' institutionalized practices are likely to differ, reflecting the reality of having different organizations working together. We adopt the construct of institutional distance to refer to these differences, defined as the "difference/similarity between the regulatory, cognitive, and normative institutions of two or more institutional environments" (Kostova, 1996:30). This construct was first proposed to represent symbolic gaps between two countries affecting the transfer of practices across organizations (Kostova, 1999). Researchers have used the construct of *country institutional profiles* based on the aggregate measures of two countries' regulative, normative and cognitive elements (Xu & Shenkar, 2002).

While organization studies rely on aggregate measures of country institutional profiles to estimate a global institutional distance, we offer a more granular conceptualization of the construct. We suggest that an institutional distance may exist for each dimension of the institutional profiles involved in an OISD project, and that it has two constitutive elements: (1) the difference between the practices related to a particular dimension of the parties' institutional profiles and (2) the pillars on which those differing practices are based. For example, two organizations may differ on the risk dimension of their project management practices. Yet, it is possible that in both cases the risk assessment practices rely on normative pillars – e.g., based on the norms of the PMI and the Australian Institute of Project Management. It is also possible that these practices rely on a normative pillar for one party and on a regulative pillar – a country law – for the other party. In such a case, not only do both organizations follow different risk assessment practices; they also rely on different institutional logics to enforce them.

2.4.3. Component 3: Conflicting Institutional Demands as Signaled by Institutional Distance

In an OISD project, the demands made by one party's institutionalized practices may conflict with those of the other party's. We define an instance of conflicting institutional demands as a situation wherein, for a given practice, actors are subjected to the demands emerging from another institutional environment that mandate a behavior different from that of their own institutional environment, as indicated by the salience of an institutional distance between those environments.

In organization studies, the notion of conflicting institutional demands has been identified in a variety of contexts (e.g., D'Aunno et al., 1991; Pache & Santos, 2010) though never formally defined. It is important to note that not all instances of institutional distance will result in conflicting institutional demands. Indeed, when the differences between institutionalized practices are not salient during a given phase or for a given activity, we posit that institutional distance does not yield conflicting institutional demands. It is when parties realize that their respective institutionalized practices are different – the *salience* of institutional distance – and that the demands of those practices are contradictory that conflicting institutional demands emerge. This can occur because parties rely on different practices to perform a given task (e.g., risk assessment) or because there is no practice institutionalized by one of the parties to perform that task.

To illustrate, we refer to the case of a vendor working for a client where many rules governing OISD projects exist. One such example is NASA, where those rules are stated in documents such as the Software Engineering Requirements document (NASA, 2014) which contains a number of sections (e.g., risk management, software measurement) outlining the activities a vendor must undertake when working for the government agency. One of those sections relates to risk management. If a vendor's risk management practices are based on a different set of norms than NASA, they may realize that their practices differ and the institutional distance between their respective risk management practices is salient. However, the vendor may not realize that the means employed to communicate the results of performing risk management with NASA differ until they have to communicate those results. In this case, the institutional distance between the communication dimensions of parties' institutionalized practices only becomes salient once the practice must be enacted.

2.4.4. Component 4: Strategic Responses as Mechanisms to Address Instances of Conflicting Institutional Demands

Arguing that compliance is not the only course of action available in the face of institutional pressures, Oliver (1991) proposed five strategic responses that actors may enact, each associated with a series of tactics summarized in Table 2.3. These strategic responses were originally proposed as mechanisms to address instances of conflicting institutional demands within a unique institutional environment defined primarily by regulative and, to a lesser extent, normative institutions (Oliver, 1991:147). Consistent with the notion of institutional pluralism (Kraatz & Block, 2008) placing organizations at the crossroads of multiple, sometimes competing, institutional environments, recent works have extended Oliver's work and suggested that the strategic responses she identified also apply in situations where conflicting institutional demands emerge from an organization's belonging to multiple institutional environments. Indeed, it has been suggested that in such situations, complying with one institutional environment's demands may be done at the expense of the demands from the other institutional environment (Pfeffer & Salancik, 2003:27). For instance, Pache and Santos (2010) developed a series of propositions to predict the selection of strategic responses based on the ability for institutional actors to negotiate the terms of an instance of conflicting institutional demands and the number of internal representations of that instance within the organization. We argue that within the context of an OISD project, these strategic responses may be used by parties to address instances of conflicting institutional demands.

Table 2.3. Strategic responses and associated tactics
(adapted from Oliver, 1991:152)

Strategy	**Tactics**	**Definition (from Pache & Santos, 2010)**
Acquiescence	Habit; Imitate; Comply	"Adoption of arrangements required by external institutional constituents" (p. 462)
Compromise	Balance; Pacify; Bargain	"Attempt by organizations to achieve partial conformity with all institutional expectations through the mild alteration of the demands, the mild alteration of the responses, or through a combination of the two" (p. 462)
Avoidance	Conceal; Buffer; Escape	"Attempt by organizations to preclude the necessity to conform to institutional pressures of to circumvent the conditions that make this conformity necessary" (p. 462)

Defiance	Dismiss; Challenge; Attack	"Explicit rejection of at least one of the institutional demands in an attempt to actively remove the source of contradiction" (p. 463)
Manipulation	Co-opt; Influence; Control	"Active attempt to alter the content of institutional requirements and to influence their promoters" (p. 463)

We suggest that evidence from the OISD literature can be interpreted along those five strategic responses. For example, Kappos (2009:192) reports on a case where a client mimicked their vendor's hierarchical structure in order to ensure that change requests would be acknowledged by the vendor and resolve communication issues between them [enacting *acquiescence*]. In another study, Vial (2009) observes that a vendor concealed their lack of compliance with a client's normative use of specific tools to produce use case documents by writing a piece of software to translate their own use case documents into the format required by the client [enacting *avoidance*]. In another instance in Vial (2009:121), parties reached a compromise on the use of modified code audit rules mandated by the client in order to integrate some of the enhancements proposed by the vendor [enacting *compromise*]. In their study of OISD projects between a British client and an Indian vendor, Nicholson and Sahay (2001) report on a client's use of manipulation to force the use of their structured approach by its vendor [enacting *manipulation*]. While some employees at the vendor enacted defiance by resigning from their jobs as a form of protest toward this strategy [enacting *defiance*], it eventually helped the client assert its power over the vendor.

2.4.5. Explaining the Enactment of Strategic Responses to Conflicting Institutional Demands in OISD Projects

In this section, we offer an explanation of the enactment of strategic responses to conflicting institutional demands. Recall that, as illustrated in Figure 2, we contend that conflicting institutional demands occur when the institutional distance between two parties' institutional profiles is salient. As synthesized in Table 2.4, we propose that the two constitutive elements of institutional distance – the difference between practices and the party with the prevailing institutional pillar – help explain which strategic response a party enacts when faced with conflicting institutional demands. We also offer a third element – sightedness – to enrich our explanation. For the sake of providing a synthetic representation, we use Boolean operators to refer to the various conditions that can explain the enactment of a given strategic response. The intention is not to offer a configurational-analysis (Rihoux & Ragin, 2009) type of explanation.

Table 2.4. Understanding strategic responses to conflicting institutional demands

Strategic response enacted by party A	Salient institutional distance			Operator	Sightedness of party A
	Difference between practices	Operator	Party with prevailing institutional pillar		
ACQUIESCENCE	Low	OR	B	N/A	N/A
COMPROMISE	High	AND	Ø	AND	Long term
AVOIDANCE	High	AND	B	AND	Short term
DEFIANCE	High	AND	A	AND	Short term
MANIPULATION	High	AND	A	AND	Long term

The first constitutive element of institutional distance, the *difference between practices,* refers to the degree of unlikeness of the practices related to a particular dimension of the parties' institutional profiles. For example, risk management practices for parties which both follow prescriptions from the PMI will probably be more alike than those followed by parties which rely on *ad hoc* practices. For the sake of parsimony, we propose that the difference between practices involved in an instance of conflicting institutional demands may be defined as either high or low.

The second constitutive element of institutional distance, the *party with the prevailing institutional pillar,* refers to the party whose practice of interest is based on the pillar that exerts the strongest pressure for compliance. Here we draw from the seminal role of legitimacy in institutional theory to explain this notion, following recent institutional works that have suggested that institutional pillars be part of the analysis of conflicting institutional demands (Pache & Santos, 2010). Institutional pillars exert varying levels of pressures based on three factors (Scott, 2008): (1) the sense of "obligation" they create based on the possibility for others to monitor compliance with a prescribed behavior; (2) the degree of "precision" of those behaviors prescribed by the institution; and (3) the possibility for "delegation" of the enforcement of the institution to third parties. The example of NASA (NASA, 2014) provides an illustration of the ability to describe and enforce regulative and normative institutions more easily than cultural-cognitive institutions. Accordingly, we propose that regulative institutions exert the strongest pressure to foster compliance, followed by normative institutions and cultural-cognitive institutions. We argue that the selection of strategic responses is explained in part by the choice institutional actors will make based on the legitimacies – and therefore the associated sanctions and rewards – of the practices involved in an instance of conflicting institutional demands. For example, practices founded on a normative or cultural-cognitive pillar may not be as strongly advocated as practices mandated by laws. This element results in one of three scenarios to explain a party's strategic response: (1) the pillar of their institutionalized practice prevails over that of the other party; (2) the pillar of the institutionalized practice of the other party prevails over their own; or (3) the pillars of the institutionalized practices of both parties are the same, i.e., there is no prevailing pillar.

We also suggest that the *sightedness* of the party enacting a given strategic response contributes to explain the enactment of this response. We define sightedness as whether a party focuses on a short-term, project-based vision or a long-term vision, considering the project part of a long-term relationship. Consistent with Oliver (1991), we consider that the five strategic responses involve varying levels of implication and resources from the part of the party enacting them. We propose that an important factor in explaining the choice of a strategic response which may be detrimental to a party's own legitimized institutionalized practices is embedded in their ability to envision work with the other party beyond the project at hand and therefore engage significant resources to support this vision. To illustrate this, we refer to Vial (2009) where the vendor may work against its own contractual agreement and official change request process to maintain a positive relationship with their clients: "if we realize that the client will slip and fall in a hole even if we warned them that they will need studs and we realize they don't have them, we will still lend them a hand and we will put the studs on their shoes, because from a commercial standpoint, it would be a bad… […] We will not let somebody die, that's the reality" (p. 90). The sightedness element allows us to explain a party's willingness to commit significant resources to develop long term relationships through series of OISD projects (e.g., Sabherwal, 1999).

Acquiescence. *Acquiescence* occurs in an OISD project when one party willingly follows the other party's practice. As shown in Table 2.4, acquiescence can be explained by a low degree of difference between practices, for example when both parties adhere to the prescriptions of the PMI for risk management or employ developers certified as Scrum masters. A party may also enact acquiescence even when practices are significantly different, but the other party's institutional pillar prevails. This may happen in one of two cases. First, a party's own practices may not be institutionalized and therefore be devoid of the legitimacy required to generate a perception that an instance of conflicting institutional demands exists. In such settings, the lack of institutionalized practices renders acquiescence easier. Second, acquiescence may be enacted when the other party's institutionalized practices exert a stronger pressure for compliance. In the example of the NASA (NASA, 2014), a vendor's acquiescence to U.S. laws and policies described in section 2.1 of the document at the expense of their own normative and cultural-cognitive institutions may be explained by the pressure for compliance with those institutions not being as strong as that exerted by the U.S. legal system and their associated sanctions (i.e., NASA's institutional pillar prevails over the vendor's). As the header of the NASA document states, in capitalized bold red letters, "compliance is mandatory." In terms of explaining acquiescence, we argue that sightedness is irrelevant because its enactment does not require an extensive amount of resources, especially when parties have similar practices

Compromise. In an attempt to partially comply with institutional pressures, an institutional actor can *compromise*. We argue that compromise occurs in situations where the difference between practices enacted by parties is high and there is no prevailing institutional pillar involved in the instance of conflicting institutional demands. Indeed, when parties both rely on the same institutional pillar to legitimize different practices, they may not be able to exert an amount of pressure high enough to warrant compliance by the other party. Compromise represents an attempt to foster a productive exchange in an instance of conflicting institutional demands. It involves the acknowledgement that the ambiguity raised by the institutional distance between parties' institutionalized practices may not be fully resolved at the benefit of one party. As a result, parties negotiate an outcome where the pressures exerted by the institutionalized practices involved in the instance of conflicting institutional demands are at least partially satisfied. For example, Gregory et al. (2013) report on a case where compromises on the enactment of work practices relevant for both parties provide an important means to address communication issues that result from significant differences regarding their communication practices. An interesting effect of compromise may be the enactment of hybrid practices that aim at integrating elements of each party's institutionalized practices in order to partially comply with their respective institutional demands. These hybrid practices and the exchanges they rely on provide a fertile ground for parties to learn about each other's practices. For instance, Vial (2009) provides an example where negotiations between a client and a vendor led to the modification of code audit rules mandated by the client were amended to incorporate some of the vendor's coding standards. This also allowed the vendor to learn about some of the client's code audit rules and integrate them into their own coding standards.

Beyond the high degree of difference between practices and the absence of a prevailing institutional pillar, we argue that long term sightedness is an additional explanatory element of compromise because this strategic response requires the dedication of significant resources from both parties. A party may be willing to forgo their own institutions for the short term through the enactment of acquiescence. However, the enactment of compromise only provides partial compliance with one's own institutions, thereby removing their ability to categorically forgo or enforce those institutions. Within the context of an outsourced project, this lack of definite resolution may be explained through a vision of the collaboration between parties which goes beyond the execution of the project at hand and takes into account potential for future business.

Avoidance. When a party actively tries not to satisfy the institutional demands of the other party without overtly expressing it, it enacts *avoidance*. In an OISD setting, we argue that avoidance is first explained by a high degree of difference between practices. Avoidance is an attempt to reject compliance with another party's institution and therefore requires the active engagement of a party's resources to either pretend that compliance is achieved, prevent validations which would uncover a lack of compliance, or remove itself from the influence of the other institutional environment altogether. We argue that the enactment of avoidance by a given party is further explained by their short sightedness and the prevalence of the other party's institutional pillar over their own. For example, in agile approaches, short iteration cycles rely on timely and informative feedback from users. A vendor and a client may initially define in their contractual agreement that an agile approach will be followed. In this case, the client may declare approval for every iteration of the software while in fact delaying proper acceptance tests until its final delivery as per practices within their institutional profile that are based on a waterfall approach. Due to the need to comply with an agile approach in the contract, the client may pretend to perform quality assurance simply because their own institutions do not prevail in this particular instance of conflicting institutional demands and they do not expect to conduct business with this vendor in the future. Similarly, a vendor may feign compliance with a client's requirements and hide behind symbolic representations of compliance in order to "win" a contract with them. For example, Levina and Vaast (2008:315) provide the example of clients complaining of Indian vendors relying on their CMM certifications as a façade to conceal their inability to maintain documentation and enact repeatable processes. Because they operate in a highly regulated environment, those clients' institutionalized practices prevail over vendors who rely on *ad hoc* practices that either are not institutionalized or rely on a cultural-cognitive pillar.

Defiance. When the requirements of an institutional demand are being explicitly questioned, *defiance* is enacted. In an OISD project, this translates into one party challenging the practices within the other party's institutional profile by not enacting mandated practices. As an active form of protest, defiance can threaten the course of the project and the relationship between parties. We explain a party's enactment of defiance by a high degree of difference between practices and the prevalence of that party's institutional pillar combined with short-term sightedness. Indeed, defiance is an active response mechanism which translates into overtly ignoring the pressure exerted by another party's institutions, actively challenging those institutions or attacking their underlying logic. Defiance aims at having one's own institutions prevail completely at the expense of others by questioning their legitimacy. In an OISD setting, a party may not always be in a position to enact defiance as it would put itself at risk of facing strong sanctions associated with non-compliance. For example, if a vendor adheres to agile development values, it will value "working software over comprehensive documentation" (Beedle et al., 2001) and may reject a client's mandated practice regarding the production of exhaustive documentation. Under the terms of the contractual agreement binding parties, the vendor may have little opportunity to challenge such practices if the client's practices are included in the contract and may therefore resort to acquiescence or avoidance as previously illustrated in Vial (2009). If however the vendor's practices prevail under the terms of the legal contract, it may defy a client's attempts at asking for comprehensive documentation by challenging the relevance or usefulness of such practice. Because defiance threatens the legitimacy of the other party's own institutions, we argue that it is focused on the short-term resolution of conflicting institutional demands, without regards or even at the expense of long term collaboration between parties. For instance, a party often resorting to defiance may foster a negative relationship with the other party throughout the course of the project while precluding the possibility of future business.

Manipulation. The last strategy, *manipulation* refers to active attempts to modify the logic underlying the demands of an institution through co-optation, influence and controlling tactics. In an OISD setting, the enactment of manipulation by a given party can be explained by a high degree of difference between practices and the prevalence of that party's institutional pillar combined with long sightedness. Like defiance, manipulation is founded upon the questioning of the underlying logic of a party's institution. However, unlike defiance, manipulation focuses on covert attempts to change this logic and "shape and redefine institutionalized norms and external criteria of evaluation, and to control or dominate the source, allocation or expression of social approval and legitimation" (Oliver, 1991:159). Given the relative stability of institutions and their legitimacy within an institutional field, manipulation represents an attempt to alter the other party's institutional logic and challenge a system which has been taken for granted for a significant amount of time. Therefore, we argue that a party must not only have the opportunity to have its own institutional pillar prevail but also envision their collaboration with the other party beyond the project at hand. Indeed, one may otherwise decide to use defiance as a strategic response because their own institutional pillar provides a stronger basis for legitimacy in a particular instance of conflicting institutional demands. However, driven by the desire to alter the fabric of the other party's own institutions over the long run (e.g., to foster synergy between their system development or project management processes), one may instead resort to manipulation, as previously illustrated in Nicholson and Sahay (2001) where a client used their position as a client with normative and regulative institutionalized practices to influence its vendors into adopting their own development approach as it sought to develop long term partnerships with them and rely on a standardized approach to execute all of their OISD projects.

From an institutional perspective, OISD projects represent a context that is prone to the emergence of conflicting institutional demands. For a given instance of conflicting institutional demands, we have argued that the analysis of the difference between practices, the prevailing institutional pillar, and the sightedness of the party enacting a given strategic response explain the selection of those responses.

2.4.6. The Enactment of Strategic Responses in an OISD Project

In this section, we provide examples of the enactment of strategic responses to illustrate the *versatility* of our explanation that is, “the degree to which it can encompass a broad domain of developmental patterns without modification of its essential character” (Poole et al., 2000:43). For the sake of simplicity we provide illustrations during two phases covering the life cycle of a project: (1) contractual negotiations and; (2) the execution of the work. These are consistent with typical project breakdowns (e.g., Ali Babar et al., 2007; Sabherwal, 2003). Also, although any response can be enacted during either phase, we limit our illustration to a subset of responses for each phase.

Contractual negotiations. During contractual negotiations, parties engage in a series of exchanges to determine the terms of the project. During that phase, a party may *acquiesce* to practices that are in line with the other party’s institutionalized practices. For example, a client may not possess much technical expertise related to software development if they seldom engage in in-house ISD. In such a case, practices related to *coding* will be absent from their institutional profile. A vendor’s institutional profile, on the other hand, may contain normative or cultural-cognitive institutions related to coding as it is a routine operation for them. While practices related to the *coding* dimensions of their institutional profiles may differ, the vendor’s institutional pillar related to that dimension prevails over the client’s. Together, these elements explain why a client acquiesces to follow the vendor’s institutions regarding the coding of the artifact.

Conversely, a party may enact *defiance* if they refuse to comply with the other party’s institution. For example, the laws in the vendor’s country may include strict penalties for overtime and therefore represent regulative institutions for the *human resources* dimension of their institutional profile. Conversely, the labor laws in the client’s home country may be more loosely defined, thereby failing to provide clear rules for actors to follow as regulative institutions. As a result, a client may have institutionalized human resources practices based on a normative pillar and focus on “getting the job done”. During contractual negotiations, the vendor’s enactment of defiance is explained by the fact that their institutional pillar (regulative) prevails over the client’s (normative) and they are more concerned with the short-term impact of the instance of conflicting institutional demands – the possibility of legal sanctions – than the long-term possibility of working with the client. To that effect, the vendor may demand that the client accept to include a contractual clause limiting the number of hours worked to a set amount in accordance with their regulative institutions.

Execution of the work. During the project a party may enact *compromise* to ensure compliance with part of their own institutions. For example, there may be an institutional distance between a client and a vendor's *communication* practices. More specifically, while both parties may enact cultural-cognitive institutions related to communication, they may vary greatly, consistent with evidence from past works on OISD detailing the influence of culture on communication practices (e.g., Gregory et al., 2013; Jain et al., 2011; Wareham et al., 2007). When these differences become salient, one party may reach out to the other to adopt hybrid communication practices which partially comply with the demands exerted by their respective institutional profiles. Indeed, in this instance of conflicting institutional demands, there is no prevailing institutional pillar and parties engage significant resources to achieve a compromise, in line with a long-term sightedness. While compromise results in the enactment of a hybrid practice by a party, it does not alter the institution in place for that party. Compromise, like acquiescence, offers the possibility for a party to willingly enact practices that differ from their own for a limited amount of time, without threatening the legitimacy of their institutions. In our example using cultural-cognitive institutions, what could be perceived as a change of culture resulting from exchanges with another party is in fact the enactment of compromise and does not have an immediate effect on the organization's institutions or culture.

2.5. Conclusion

Our work builds on extant research that portrays differences between parties as critical hinderers of OISD projects. Using institutional theory as a foundational theory, we offer a *process explanation* of the effects of those differences through the interaction of four components: (1) the parties' institutional profiles; (2) the institutional distance between parties' practices within those institutional profiles; (3) conflicting institutional demands which emerge when the differences between parties' institutionalized practices become salient; and (4) the institutional strategic responses (Oliver, 1991) that can be enacted to address instances of conflicting institutional demands. Our main contribution being *theoretical,* we use the four essential elements of theory development proposed by

Whetten (1989:490-492) – and consistent with Weber (2003) – to describe the main features of our contribution: the (1) What; (2) How; (3) Why; and (4) Who, Where, When.

The first element [What] pertains to the constructs that contribute to explaining the phenomenon of interest. Our work adds to extant literature by conceptualizing differences between parties under the lens of institutional theory. We propose the construct of institutional profiles as a set of initial conditions that parties bring into the project and carry throughout its course. In doing so, we add to past conceptualizations of differences in OISD projects by explicitly theorizing on the properties of differences to better understand their effects. We also enrich past conceptualizations by adapting the construct of institutional distance (Kostova, 1997) and proposing the notion of *salience* of institutional distance – when parties realize that their respective institutionalized practices differ – as instrumental in defining the conditions that allow for the emergence of conflicting institutional demands.

The second element [How] pertains to the relationships between the constructs that form the explanation of the phenomenon of interest. While differences between parties have typically been conceptualized as sources of issues in OISD projects, our explanation questions this direct link and includes *salience* of institutional distance as an element that transforms institutional distance into an instance of conflicting institutional demands. Our work provides new insight into the complex relationship that exists between differences and difficulties in OISD projects. Our explanation suggests that in the presence of an instance of conflicting institutional demands, it is the properties of that instance – the difference between practices and the prevailing institutional pillar – along with the sightedness of the party enacting a strategic response that together provide the necessary elements to explain the enactment of a given strategic response.

The third element [Why] deals with the “theoretical glue that welds” (Whetten, 1989:491) the explanation together. We have adopted institutional theory as a foundational theory to bridge the components of our explanation under a common set of theoretical assumptions. By focusing on institutionalized practices along with their institutional pillars, our explanation allows us to take into account all of those practices that are institutionalized within an organization, whether they relate to national, organizational, or any other level of culture or norms. Our conceptualization of differences departs significantly from the literature on OISD and outsourcing in general. Past works on OISD refer to differences as a degree of unlikeness of a given aspect for both parties (e.g., culture, ISD methodology). While this conceptualization of differences works if parties both enact a given practice based on the same logic (i.e., culture), it fails to take into account the possibility that one party may for example enact a practice based on their national culture while the other will enact a different practice because of the laws in their home country. Our explanation builds on the properties of institutionalized practices to account for the fact that practices, as well the institutional logics that drive their enactment, may differ between parties. Together these two aspects allow us to build a fine-grained explanation of the effects of differences between institutionalized parties in OISD projects based on their respective institutional pillars.

The fourth element [Who, Where, When] deals with the "boundaries" (Whetten, 1989:492) of the explanation that define its context of applicability (Rivard, 2014; Weber, 2003). Based on institutional theory's conceptual assumptions, our model is also bounded by five contextual assumptions. Together, these assumptions represent the main limitations of our explanation, which: (1) focuses on the effect of differences between institutionalized practices in OISD projects; (2) in line with the tenets of institutional theory, assumes that institutions will remain stable for the duration of a project; (3) is limited to OISD projects involving only two parties; (4) pertains to ISD projects where a software artifact is built, configured, or maintained; and (5) assumes that the project requires regular exchanges between parties over its course.

Although they limit our explanation, these boundaries provide the framing that fosters the arguments that together define the four components of our explanation as well as their interaction. Notwithstanding, our explanation exhibits *versatility* (Poole et al., 2000) through its applicability throughout an OISD project in a consistent manner, whether during contractual negotiations or the execution of work.

While conceptual in nature, our work has implications for practitioners. First, the construct of institutional profile and its constituting dimensions provide a means for managers of OISD projects to account for a number of technical as well as managerial practices that are relevant in this context. Indeed, while practices driven by legal requirements may be easily identifiable because they are crystallized in official rules and regulations, our work suggests that practices relying on a normative and cultural-cognitive pillar are important to explain the enactment of strategic responses when conflicting institutional demands arise, even though they may have become so engrained in the daily activities of the organization that they may not be identified as problematic *a priori*. The quasi-hierarchy drawn by the three institutional pillars may also be of interest to practitioners seeking to identify those practices which must be enforced and those on which they may be willing to compromise with another party.

Second, the inclusion of salience as an explicit element of our explanation invites practitioners to consider differences with their OISD partners based on their relevance within the context of a given project. For instance, cultural differences may only be relevant for certain aspects of an OISD project based on the dimension of the parties' institutional profiles they relate to. As a result, managers may forgo certain strategies (e.g., have all vendor staff come for a client site visit) while in other cases target specific actors (e.g., project management staff) at specific points in time. In addition, our explanation posits that a number of differences may be identified during contractual negotiations, leaving more time for practitioners to be cognizant of those differences, the effects they may have on the project and the strategies they may enact to address them.

Building on our explanation, future research could develop other, related explanations focused on the impacts of differences on the project or relationship between parties over time. Indeed, our work is a first toward the development of process theories (Poole et al., 2000) that conceptualize OISD projects as sequences of events unfolding over time. One such avenue is the study of *patterns* of strategic responses enacted at various points in time – both within and across projects – to explain the use of formal or relational governance mechanisms. For example, the repeated enactment of compromise during contractual negotiations may explain why parties rarely refer to the written contract during the course of the project. Conversely, the enactment of defiance and avoidance may hinder the relationship between parties and explain why they refer more frequently to formal control mechanisms. A second avenue is the study of the enactment of strategic responses as opportunities to trigger processes of deinstitutionalization and reinstitutionalization in an organization. One of our boundary conditions, in line with core assumptions of institutional theory, assumes that the institutional profiles of parties – their institutionalized practices – are expected to remain stable for the duration of the project. However the enactment of strategic responses such as acquiescence, compromise or manipulation seeks to modify, in part or in whole, a party's institutions. Future research may study how these strategic responses offer an opportunity for a party to challenge the legitimacy of their own institutions and learn from the other party, thereby altering the *institutional scripts* (Barley & Tolbert, 1997) in place and explain the initiation of a process of deinstitutionalization, or the evolution of institutions across a series of projects between parties.

References

ABDULLAH LM and VERNER JM (2012) Analysis and application of an outsourcing risk framework. *The Journal of Systems and Software, 85*(8), 1930-1952.

AHUJA MK and SINCLAIR RF (2012) The influence of outsourcing models on vendor knowledge integration. *Journal of Information Technology Theory and Application, 13*(4), 5-19.

ALI BABAR M, VERNER JM and NGUYEN PT (2007) Establishing and maintaining trust in software outsourcing relationships: An empirical investigation. *Journal of Systems and Software, 80*(9), 1438-1449.

ANG S and CUMMINGS LL (1997) Strategic response to institutional influences on information systems outsourcing. *Organization Science, 8*(3), 235-256.

ARNOTT D, JIRACHIEFPATTANA W and O'DONNELL P (2007) Executive information systems development in an emerging economy. *Decision Support Systems, 42*(4), 2078-2084.

AVGEROU C (2000) IT and organizational change: An institutionalist perspective. *Information Technology & People, 13*(4), 234-262.

AVISON D and FITZGERALD G (2006) *Information systems development: Methodologies, techniques & tools*. McGraw-Hill Education, New York, NY.

AYRES BJ (2003) *Institutional influences and control of software development projects: An examination of air force software project teams.* PhD dissertation, Florida State University, Tallahassee, FL.

BARCA C and CORDELLA A (2006) Seconds out, round two: Contextualizing e-government projects within their institutional milieu: A London local authority case. *Scandinavian Journal of Information Systems, 18*(1), 37-60.

BARLEY SR and TOLBERT PS (1997) Institutionalization and structuration: Studying the links between action and institution. *Organization Studies, 18*(1), 93-117.

BECK K and ANDRES C (2004) *Extreme programming explained: Embrace change*. Addison-Wesley Professional, Boston, MA.

BECK R (2014) 'Looking for trouble' in global information systems development and new product development outsourcing projects. In *Innovation and IT in an international context: R&D strategy and operations* (ROWE F and TE'ENI D, Eds), pp. 236-248, Palgrave Macmillan, New York, NY.

BEEDLE M, et al. (2001) Manifesto for agile software development. http://agilemanifesto.org/, accessed March 15, 2013.

BELLAH JC, BURNS JR and CASSIDY CM (2013) Offshore information systems development process in India: How practitioners respond to the challenges. *Journal of Information Technology Case and Application Research, 15*(2), 30-53.

BENITO GRG, DOVGAN O, PETERSEN B and WELCH LS (2013) Offshore outsourcing: A dynamic, operation mode perspective. *Industrial Marketing Management, 42*(2), 211-222.

BHAT JM, GUPTA M and MURTHY SN (2006) Overcoming requirements engineering challenges: Lessons from offshore outsourcing. *IEEE Software, 23*(5), 38-44.

BURMISTROV I (2006) A new destination for offshore usability. *Interactions, 13*(2), 22-24.

CARMEL E and AGARWAL R (2001) Tactical approaches for alleviating distance in global software development. *IEEE Software, 18*(2), 22-29.

CARPENTER VL and FEROZ EH (2001) Institutional theory and accounting rule choice: An analysis of four US state governments' decisions to adopt generally accepted accounting principles. *Accounting, Organizations and Society, 26*(7), 565-596.

CHOUDHURY V and SABHERWAL R (2003) Portfolios of control in outsourced software development projects. *Information Systems Research, 14*(3), 291-314.

CUSICK J and PRASAD A (2006) A practical management and engineering approach to offshore collaboration. *IEEE Software, 23*(5), 20-29.

D'AUNNO T, SUTTON RI and PRICE RH (1991) Isomorphism and external support in conflicting institutional environments: A study of drug abuse treatment units. *The Academy of Management Journal, 34*(3), 636-661.

DAVEY H and ALLGOOD B (2002) Offshore development, building relationships across international boundaries: A case study. *Information Strategy: The executive's journal, 18*(3), 13-16.

DIBBERN J, WINKLER J and HEINZL A (2008) Explaining variations in client extra costs between software projects offshored to India. *MIS Quarterly, 32*(2), 333-366.

DUBÉ L and ROBEY D (1999) Software stories: Three cultural perspectives on the organizational practices of software development. *Accounting, Management and Information Technologies, 9*(4), 223-259.

FELDMAN MS and ORLIKOWSKI WJ (2011) Theorizing practice and practicing theory. *Organization Science, 22*(5), 1240-1253.

FELDMAN MS and PENTLAND BT (2003) Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly, 48*(1), 94-118.

FITZGERALD B, RUSSO NL and STOLTERMAN E (2002) *Information systems development: Methods in action*. McGraw-Hill Education, New York, NY.

FRANK SJ (2005) Source out, risk in. *IEEE Spectrum, 42*(4), 60-62.

GONZALEZ R, GASCO J and LLOPIS J (2010) Information systems offshore outsourcing: An exploratory study of motivations and risks in large spanish firms. *Information Systems Management, 27*(4), 340-355.

GOPAL A, ESPINOSA JA, GOSAIN S and DARCY DP (2011) Coordination and performance in global software service delivery: The vendor's perspective. *IEEE Transactions on Engineering Management, 58*(4), 772-785.

GOPAL A and GOSAIN S (2010) The role of organizational controls and boundary spanning in software development outsourcing: Implications for project performance. *Information Systems Research, 21*(4), 960-982.

GOPAL A and KOKA BR (2010) The role of contracts on quality and returns to quality in offshore software development outsourcing. *Decision Sciences, 41*(3), 491-516.

GOPAL A and KOKA BR (2012) The asymmetric benefits of relational flexibility: Evidence from software development outsourcing. *MIS Quarterly, 36*(2), 553-576.

GOPAL A, MUKHOPADHYAY T and KRISHNAN MS (2002) The role of software processes and communication in offshore software development. *Communications of the ACM, 45*(4), 193-200.

GOWAN JA, JR. and MATHIEU RG (2005) The importance of management practices in IS project performance: An empirical study. *Journal of Enterprise Information Management, 18*(1/2), 235-255.

GREENWOOD R, SUDDABY R and HININGS CR (2002) Theorizing change: The role of professional associations in the transformation of institutionalized fields. *Academy of Management Journal, 45*(1), 58-80.

GREGOR S (2006) The nature of theory in information systems. *MIS Quarterly, 30*(3), 611-642.

GREGORY RW, BECK R and KEIL M (2013) Control balancing in information systems development offshoring projects. *MIS Quarterly, 37*(4), 1211-1232.

GREGORY RW, PRIFLING M and BECK R (2009) The role of cultural intelligence for the emergence of negotiated culture in IT offshore outsourcing projects. *Information Technology & People, 22*(3), 223-241.

HERBSLEB JD, PAULISH DJ and BASS M (2005) Global software development at Siemens: Experience from nine projects. In *Proceedings of the 27th International Conference on Software Engineering* (ROMAN GC, GRISWOLD W and NUSEIBEH B, Eds), pp. 524-533, St. Louis, MO.

HOFSTEDE G, NEUIJEN B, OHAYV DD and SANDERS G (1990) Measuring organizational cultures: A qualitative and quantitative study across twenty cases. *Administrative science quarterly, 35*(2), 286-316.

HUANG H and TRAUTH EM (2007) Cultural influences and globally distributed information systems development: Experiences from chinese IT professionals. In *Proceedings of the 2007 ACM SIGMIS CPR conference on computer personnel research* (LENDING D and VICIAN C, Eds), pp. 36-45, St. Louis, MO.

HUBER TL, FISCHER TA, DIBBERN J and HIRSCHHEIM R (2013) A process model of complementarity and substitution of contractual and relational governance in IS outsourcing. *Journal of Management Information Systems, 30*(3), 81-114.

IACOVOU CL and NAKATSU R (2008) A risk profile of offshore-outsourced development projects. *Communications of the ACM, 51*(6), 89-94.

ISLAM S, JOARDER MMA and HOUMB SH (2009) Goal and risk factors in offshore outsourced software development from vendor's viewpoint. In *Proceedings of the Fourth IEEE Conference on Global Software Engineering* (VERNER JM and PAULISH DJ, Eds), pp. 347-352, Limerick, Ireland.

JACCARD J and JACOBY J (2009) *Theory construction and model-building skills: A practical guide for social scientists*. Guilford Press, New York, NY.

JAIN RP, SIMON JC and POSTON RS (2011) Mitigating vendor silence in offshore outsourcing: An empirical investigation. *Journal of Management Information Systems, 27*(4), 261-298.

JARVENPAA SL and MAO JY (2008) Operational capabilities development in mediated offshore software services models. *Journal of Information Technology, 23*(1), 3-17.

JEPPERSON RL (1991) Institutions, institutional effects, and institutionalism. In *The new institutionalism in organizational analysis* (POWELL WW and DIMAGGIO P, Eds), pp. 143-163, University of Chicago Press, Chicago, IL.

KANNABIRAN G and SANKARAN K (2011) Determinants of software quality in offshore development - an empirical study of an Indian vendor. *Information and Software Technology, 53*(11), 1199-1208.

KAPPOS A (2009) *Towards an understanding and an explanation of the relationship between culture and IS.* PhD dissertation, HEC Montreal, Montreal, Canada.

KEIL M, IM GP and MÄHRING M (2007) Reporting bad news on software projects: The effects of culturally constituted views of face-saving. *Information Systems Journal, 17*(1), 59-87.

KHAN SU, NIAZI M and AHMAD R (2011) Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. *Information and Software Technology, 53*(7), 693-706.

KOH C, ANG S and STRAUB DW (2004) IT outsourcing success: A psychological contract perspective. *Information Systems Research, 15*(4), 356-373.

KOSTOVA T (1996) *Success of the transnational transfer of organizational practices within multinational companies.* PhD dissertation, University of Minnesota, Minneapolis, MN.

KOSTOVA T (1997) Country institutional profiles: Concept and measurement. *Academy of Management Proceedings, 1*, 180-184.

KOSTOVA T (1999) Transnational transfer of strategic organizational practices: A contextual perspective. *Academy of Management Review, 24*(2), 308-324.

KRAATZ MS and BLOCK ES (2008) Organizational implications of institutional pluralism. In *The sage handbook of organizational institutionalism* (GREENWOOD R, OLIVER C, SAHLIN K and SUDDABY R, Eds), pp. 243-275, Sage Publications, London, U.K.

LEVINA N and VAAST E (2008) Innovating or doing as told? Status differences and overlapping boundaries in offshore collaboration. *MIS Quarterly, 32*(2), 307-332.

MAHNKE V, WAREHAM J and BJORN-ANDERSEN N (2008) Offshore middlemen: Transnational intermediation in technology sourcing. *Journal of Information Technology, 23*(1), 18-30.

MARKUS ML and SAUNDERS C (2007) Editor's comments: Looking for a few good concepts...And theories...For the information systems field. *MIS Quarterly, 31*(1), iii-vi.

MATHEW SK (2011) Mitigation of risks due to service provider behavior in offshore software development. *Strategic Outsourcing: An International Journal, 4*(2), 179-200.

MIGNERAT M and RIVARD S (2009) Positioning the institutional perspective in information systems research. *Journal of Information Technology, 24*(4), 369-391.

MIGNERAT M and RIVARD S (2012) The institutionalization of information system project management practices. *Information and Organization, 22*(2), 125-153.

MIRANI R (2007) Procedural coordination and offshored software tasks: Lessons from two case studies. *Information & Management, 44*(2), 216-230.

NAKATSU RT and IACOVOU CL (2009) A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel delphi study. *Information & Management, 46*(1), 57-68.

NASA (2014) Npr 7150.2b — NASA software engineering requirements. http://nodis3.gsfc.nasa.gov/npg_img/N_PR_7150_002B_/N_PR_7150_002B_.pdf, accessed January 24, 2016.

NATOVICH J (2003) Vendor related risks in IT development: A chronology of an outsourced project failure. *Technology Analysis & Strategic Management, 15*(4), 409-419.

NICHOLSON B and SAHAY S (2001) Some political and cultural issues in the globalisation of software development: Case experience from britain and India. *Information and Organization, 11*(1), 25-43.

NURMI A, HALLIKAINEN P and ROSSI M (2005) Coordination of outsourced information system development in multiple customer environment - a case study of a joint information system development project. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 260a, Hawaii, HI.

OLIVER C (1991) Strategic responses to institutional processes. *Academy of Management Review, 16*(1), 145-179.

OZA N, HALL T, RAINER A and GREY S (2004) Critical factors in software outsourcing: A pilot study. In *Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research* (MEHANDJIEV N and BRERETON P, Eds), pp. 67-71, Newport Beach, CA.

PACHE A-C and SANTOS F (2010) When worlds collide: The internal dynamics of organizational responses to conflicting institutional demands. *Academy of Management Review, 35*(3), 455-476.

PALVIA S (2008) Challenges for small enterprises in the sourcing life cycle: Evidence from offshoring to India. *Journal of Information Technology Case and Application Research, 10*(4), 75-84.

PFEFFER J and SALANCIK GR (2003) *The external control of organizations: A resource dependence perspective*. Stanford University Press, Stanford, CA.

POOLE MS, VAN DE VEN AH, DOOLEY K and HOLMES ME (2000) *Organizational change and innovation processes: Theory and methods for research*. Oxford University Press, USA.

PRIES-HEJE J, BASKERVILLE R and HANSEN GI (2005) Strategy models for enabling offshore outsourcing: Russian short-cycle-time software development. *Information Technology for Development, 11*(1), 5-30.

PROJECT MANAGEMENT INSTITUTE (2013) *A guide to the project management body of knowledge: Pmbok guide*. Project Management Institute.

RAI A, MARUPING LM and VENKATESH V (2009) Offshore information systems project success: The role of social embeddedness and cultural characteristics. *MIS Quarterly, 33*(3), 617-A617.

RIHOUX B and RAGIN CC (2009) *Configurational comparative methods: Qualitative comparative analysis (qca) and related techniques*. Sage, Thousand Oaks, CA.

RIVARD S (2014) Editor's comments: The ions of theory construction. *Management Information Systems Quarterly, 38*(2), iii-xiii.

ROWE F (2012) Toward a richer diversity of genres in information systems research: New categorization and guidelines. *European Journal of Information Systems, 21*(5), 469-478.

ROWE F (2014) What literature review is not: Diversity, boundaries and recommendations. *European Journal of Information Systems, 23*(3), 241-255.

ROY V and AUBERT BA (2002) A resource-based analysis of IT sourcing. *ACM SIGMIS Database, 33*(2), 29-40.

SABHERWAL R (1999) The role of trust in outsourced IS development projects. *Communications of the ACM, 42*(2), 80-86.

SABHERWAL R (2003) The evolution of coordination in outsourced software development projects: A comparison of client and vendor perspectives. *Information and Organization, 13*(3), 153-202.

SCOTT WR (2008) *Institutions and organizations: Ideas and interests*. Sage Publications, Thousand Oaks, CA.

SHENKAR O, LUO Y and YEHESKEL O (2008) From “distance” to “friction”: Substituting metaphors and redirecting intercultural research. *Academy of Management Review, 33*(4), 905-923.

SUCHMAN MC (1995) Managing legitimacy: Strategic and institutional approaches. *Academy of Management Review, 20*(3), 571-610.

TEO H-H, WEI K-K and BENBASAT I (2003) Predicting intention to adopt interorganizational linkages: An institutional perspective. *MIS Quarterly*, 19-49.

TIWANA A (2004) Beyond the black box: Knowledge overlaps in software outsourcing. *IEEE Software, 21*(5), 51-58.

TIWANA A (2010) Systems development ambidexterity: Explaining the complementary and substitutive roles of formal and informal controls. *Journal of Management Information Systems, 27*(2), 87-126.

TIWANA A and KEIL M (2009) Control in internal and outsourced software projects. *Journal of Management Information Systems, 26*(3), 9-44.

TORKAR R, MINOVES P and GARRIGÓS J (2011) Adopting free/libre/open source software practices, techniques and methods for industrial use. *Journal of the Association for Information Systems, 12*(1), 88-122.

VERNER JM and ABDULLAH LM (2012) Exploratory case study research: Outsourced project failure. *Information and Software Technology, 54*(8), 866-886.

VIAL G (2009) *An institutional analysis of outsourced information systems development projects.* M. Sc. dissertation, HEC Montréal, Montreal, Canada.

WALSHAM G (2002) Cross-cultural software production and use: A structurational analysis. *MIS Quarterly, 26*(4), 359-380.

WAREHAM J, MAHNKE V, PETERS S and BJORN-ANDERSEN N (2007) Communication metaphors-in-use: Technical communication and offshore systems development. *IEEE Transactions on Professional Communication, 50*(2), 93-108.

WEBER R (2003) Editor's comment: Theoretically speaking. *MIS Quarterly, 27*(3), iii-xii.

WEBSTER J and WATSON RT (2002) Analyzing the past to prepare the future. *MIS Quarterly, 26*(2), 13-23.

WHETTEN DA (1989) What constitutes a theoretical contribution? *Academy of Management Review, 14*(4), 490-495.

WINKLER JK, DIBBERN J and HEINZL A (2008) The impact of cultural differences in offshore outsourcing - case study results from German-Indian application development projects. *Information Systems Frontiers, 10*(2), 243-258.

XU D and SHENKAR O (2002) Institutional distance and the multinational enterprise. *Academy of Management Review, 27*(4), 608-618.

ZATOLYUK S and ALLGOOD B (2004) Evaluating a country for offshore outsourcing: Software development providers in the ukraine. *Information Systems Management, 21*(3), 28-33.

Appendix 2.1: Concept Matrix of the Literature on OISD Based on the Notion of Differences and their Associated Impacts

Table A2.1.1. Differences and their associated impacts (in alphabetical order)

Type of difference	Negative impact(s)	Source(s)
Different client-representative/project leader cultures	Costs Satisfaction	Rai et al. (2009)
Different currencies	Costs	Nakatsu and Iacovou (2009)
Different geopolitical environments	Costs	Gonzalez et al. (2010)
	Delays	Natovich (2003)
	Unspecified	Nakatsu and Iacovou (2009)
Different infrastructures	Communication	Nakatsu and Iacovou (2009); Pries-Heje et al. (2005); Zatolyuk and Allgood (2004)
	Costs	Gonzalez et al. (2010)
	Delays	Bellah et al. (2013); Islam et al. (2009)
Different interests	Conflicts	Bhat et al. (2006)
	Control	Mirani (2007)
	Coordination	Sabherwal (2003)
	Costs	Natovich (2003)
	Delays	Abdullah and Verner (2012); Bhat et al. (2006)
	System Quality	Abdullah and Verner (2012); Gopal and Koka (2010)
	Trust	Sabherwal (1999)
Different languages	Communication	Ali Babar et al. (2007); Bellah et al. (2013); Bhat et al. (2006); Davey and Allgood (2002); Huang and Trauth (2007); Huang and Trauth (2007); Iacovou and Nakatsu (2008); Islam et al. (2009); Khan et al. (2011); Levina and Vaast (2008); Nakatsu and Iacovou (2009); Oza et al. (2004); Pries-Heje et al. (2005)
	Coordination	Islam et al. (2009); Sabherwal (2003)
	Costs	Gonzalez et al. (2010)
	Knowledge transfer & integration	Levina and Vaast (2008)
	Project success	Arnott et al. (2007)
Different legal environments	Control	Bellah et al. (2013)
	Costs	Jarvenpaa and Mao (2008); Mathew (2011); Khan et al. (2011)
	Delays	Frank (2005); Mathew (2011); Nakatsu and Iacovou (2009); Pries-Heje et al. (2005)
Different locations	Communication	Abdullah and Verner (2012); Bellah et al. (2013); Bhat et al.

		(2006); Burmistrov (2006); Iacovou and Nakatsu (2008)
	Control	Burmistrov (2006)
	Coordination	Abdullah and Verner (2012); Bellah et al. (2013)
	Costs	Dibbern et al. (2008); Iacovou and Nakatsu (2008)
	Delays	Cusick and Prasad (2006); Huang and Trauth (2007)
	Social relations	Nicholson and Sahay (2001)
	Trust	Bhat et al. (2006)
Different maturity levels	Communication	Wareham et al. (2007)
Different methodologies	Delays	Abdullah and Verner (2012); Bhat et al. (2006)
	System Quality	Abdullah and Verner (2012); Bhat et al. (2006)
Different national cultures	Communication	Ali Babar et al. (2007); Bellah et al. (2013); Bhat et al. (2006); Carmel and Agarwal (2001); Carmel and Agarwal (2001); Cusick and Prasad (2006); Davey and Allgood (2002); Gregory et al. (2009); Herbsleb et al. (2005); Huang and Trauth (2007); Iacovou and Nakatsu (2008); Islam et al. (2009); Jain et al. (2011); Keil et al. (2007); Khan et al. (2011); Levina and Vaast (2008); Mahnke et al. (2008); Nakatsu and Iacovou (2009); Oza et al. (2004); Wareham et al. (2007)
	Conflicts	Winkler et al. (2008); Dubé and Robey (1999)
	Control	Dibbern et al. (2008); Jain et al. (2011)
	Cooperation	Winkler et al. (2008)
	Coordination	Burmistrov (2006); Dibbern et al. (2008); Islam et al. (2009); Mahnke et al. (2008); Sabherwal (2003)
	Costs	Dibbern et al. (2008); Dubé and Robey (1999); Mahnke et al. (2008); Gonzalez et al. (2010)
	Delays	Bellah et al. (2013)
	Frustration	Herbsleb et al. (2005)
	Performance	Winkler et al. (2008)
	Satisfaction	Gregory et al. (2009)
	System Quality	Jarvenpaa and Mao (2008)
	Trust	Winkler et al. (2008)
Different organizational cultures	Communication	Herbsleb et al. (2005); Kannabiran and Sankaran (2011); Oza et al. (2004)
	Conflict	Nicholson and Sahay (2001)
	Coordination	Gregory et al. (2009)
	Frustration	Herbsleb et al. (2005)

	System Quality	Gregory et al. (2009)
Different organizations	Communication	Carmel and Agarwal (2001); Gopal et al. (2011); Iacovou and Nakatsu (2008); Islam et al. (2009)
	Control	Bellah et al. (2013); Choudhury and Sabherwal (2003); Cusick and Prasad (2006); Gopal and Gosain (2010); Khan et al. (2011); Mirani (2007); Oza et al. (2004); Tiwana (2010); Verner and Abdullah (2012)
	Cooperation	Roy and Aubert (2002)
	Coordination	Benito et al. (2013); Carmel and Agarwal (2001); Dibbern et al. (2008); Islam et al. (2009); Levina and Vaast (2008); Sabherwal (2003)
	Costs	Cusick and Prasad (2006); Gopal and Koka (2012); Iacovou and Nakatsu (2008); Tiwana (2004); Tiwana (2010)
	Delays	Choudhury and Sabherwal (2003); Nurmi et al. (2005); Palvia (2008)
	Knowledge transfer & integration	Ahuja and Sinclair (2012); Bellah et al. (2013); Dibbern et al. (2008); Gregory et al. (2009); Gregory et al. (2013); Islam et al. (2009); Jain et al. (2011); Kannabiran and Sankaran (2011); Levina and Vaast (2008); Mirani (2007); Nakatsu and Iacovou (2009); Tiwana (2004)
	Monitoring	Oza et al. (2004)
	System Quality	Choudhury and Sabherwal (2003); Iacovou and Nakatsu (2008); Islam et al. (2009); Kannabiran and Sankaran (2011); Khan et al. (2011); Palvia (2008); Sabherwal (2003)
	Project cancellation	Roy and Aubert (2002)
	Rework	Choudhury and Sabherwal (2003); Gopal et al. (2002)
	Trust	Palvia (2008)
Different professional cultures	Communication	Herbsleb et al. (2005)
	Coordination	Huang and Trauth (2007)
	Frustration	Herbsleb et al. (2005)
Different technologies	Delays	Nurmi et al. (2005)
Different time zones	Communication	Bellah et al. (2013); Carmel and Agarwal (2001); Kannabiran and Sankaran (2011); Nakatsu and Iacovou (2009)
	Coordination	Bellah et al. (2013); Carmel and Agarwal (2001); Herbsleb et al. (2005)
	Costs	Gonzalez et al. (2010)
	Delays	Cusick and Prasad (2006); Gopal et al. (2011)

	Social relations	Nicholson and Sahay (2001)
Different tools	Conflicts	Bhat et al. (2006)
Different work cultures	Complaints	Bhat et al. (2006)
	Costs	Dubé and Robey (1999)
	Escalation	Bhat et al. (2006)
Different work practices	Communication	Gregory et al. (2013)
	Control	Mahnke et al. (2008)
	Coordination	Gregory et al. (2009); Rai et al. (2009)
	Costs	Benito et al. (2013)
	Delays	Gowan and Mathieu (2005); Natovich (2003)
	System Quality	Gregory et al. (2013); Mahnke et al. (2008)
	Unspecified	Nakatsu and Iacovou (2009)

References

ABDULLAH LM and VERNER JM (2012) Analysis and application of an outsourcing risk framework. *The Journal of Systems and Software, 85*(8), 1930-1952.

AHUJA MK and SINCLAIR RF (2012) The influence of outsourcing models on vendor knowledge integration. *Journal of Information Technology Theory and Application, 13*(4), 5-19.

ALI BABAR M, VERNER JM and NGUYEN PT (2007) Establishing and maintaining trust in software outsourcing relationships: An empirical investigation. *Journal of Systems and Software, 80*(9), 1438-1449.

ARNOTT D, JIRACHIEFPATTANA W and O'DONNELL P (2007) Executive information systems development in an emerging economy. *Decision Support Systems, 42*(4), 2078-2084.

BELLAH JC, BURNS JR and CASSIDY CM (2013) Offshore information systems development process in India: How practitioners respond to the challenges. *Journal of Information Technology Case and Application Research, 15*(2), 30-53.

BENITO GRG, DOVGAN O, PETERSEN B and WELCH LS (2013) Offshore outsourcing: A dynamic, operation mode perspective. *Industrial Marketing Management, 42*(2), 211-222.

BHAT JM, GUPTA M and MURTHY SN (2006) Overcoming requirements engineering challenges: Lessons from offshore outsourcing. *IEEE Software, 23*(5), 38-44.

BURMISTROV I (2006) A new destination for offshore usability. *Interactions, 13*(2), 22-24.

CARMEL E and AGARWAL R (2001) Tactical approaches for alleviating distance in global software development. *IEEE Software, 18*(2), 22-29.

CHOUDHURY V and SABHERWAL R (2003) Portfolios of control in outsourced software development projects. *Information Systems Research, 14*(3), 291-314.

CUSICK J and PRASAD A (2006) A practical management and engineering approach to offshore collaboration. *IEEE Software, 23*(5), 20-29.

DAVEY H and ALLGOOD B (2002) Offshore development, building relationships across international boundaries: A case study. *Information Strategy: The executive's journal, 18*(3), 13-16.

DIBBERN J, WINKLER J and HEINZL A (2008) Explaining variations in client extra costs between software projects offshored to India. *MIS Quarterly, 32*(2), 333-366.

DUBÉ L and ROBEY D (1999) Software stories: Three cultural perspectives on the organizational practices of software development. *Accounting, Management and Information Technologies, 9*(4), 223-259.

FRANK SJ (2005) Source out, risk in. *IEEE Spectrum, 42*(4), 60-62.

GONZALEZ R, GASCO J and LLOPIS J (2010) Information systems offshore outsourcing: An exploratory study of motivations and risks in large spanish firms. *Information Systems Management, 27*(4), 340-355.

GOPAL A, ESPINOSA JA, GOSAIN S and DARCY DP (2011) Coordination and performance in global software service delivery: The vendor's perspective. *IEEE Transactions on Engineering Management, 58*(4), 772-785.

GOPAL A and GOSAIN S (2010) The role of organizational controls and boundary spanning in software development outsourcing: Implications for project performance. *Information Systems Research, 21*(4), 960-982.

GOPAL A and KOKA BR (2010) The role of contracts on quality and returns to quality in offshore software development outsourcing. *Decision Sciences, 41*(3), 491-516.

GOPAL A and KOKA BR (2012) The asymmetric benefits of relational flexibility: Evidence from software development outsourcing. *MIS Quarterly, 36*(2), 553-576.

GOPAL A, MUKHOPADHYAY T and KRISHNAN MS (2002) The role of software processes and communication in offshore software development. *Communications of the ACM, 45*(4), 193-200.

GOWAN JA, JR. and MATHIEU RG (2005) The importance of management practices in IS project performance: An empirical study. *Journal of Enterprise Information Management, 18*(1/2), 235-255.

GREGORY RW, BECK R and KEIL M (2013) Control balancing in information systems development offshoring projects. *MIS Quarterly, 37*(4), 1211-1232.

GREGORY RW, PRIFLING M and BECK R (2009) The role of cultural intelligence for the emergence of negotiated culture in IT offshore outsourcing projects. *Information Technology & People, 22*(3), 223-241.

HERBSLEB JD, PAULISH DJ and BASS M (2005) Global software development at Siemens: Experience from nine projects. In *Proceedings of the 27th International Conference on Software Engineering* (ROMAN GC, GRISWOLD W and NUSEIBEH B, Eds), pp. 524-533, St. Louis, MO.

HUANG H and TRAUTH EM (2007) Cultural influences and globally distributed information systems development: Experiences from chinese IT professionals. In *Proceedings of the 2007 ACM SIGMIS CPR conference on computer personnel research* (LENDING D and VICIAN C, Eds), pp. 36-45, St. Louis, MO.

IACOVOU CL and NAKATSU R (2008) A risk profile of offshore-outsourced development projects. *Communications of the ACM, 51*(6), 89-94.

ISLAM S, JOARDER MMA and HOUMB SH (2009) Goal and risk factors in offshore outsourced software development from vendor's viewpoint. In *Proceedings of the Fourth IEEE Conference on Global Software Engineering* (VERNER JM and PAULISH DJ, Eds), pp. 347-352, Limerick, Ireland.

JAIN RP, SIMON JC and POSTON RS (2011) Mitigating vendor silence in offshore outsourcing: An empirical investigation. *Journal of Management Information Systems, 27*(4), 261-298.

JARVENPAA SL and MAO JY (2008) Operational capabilities development in mediated offshore software services models. *Journal of Information Technology, 23*(1), 3-17.

KANNABIRAN G and SANKARAN K (2011) Determinants of software quality in offshore development - an empirical study of an Indian vendor. *Information and Software Technology, 53*(11), 1199-1208.

KEIL M, IM GP and MÄHRING M (2007) Reporting bad news on software projects: The effects of culturally constituted views of face-saving. *Information Systems Journal, 17*(1), 59-87.

KHAN SU, NIAZI M and AHMAD R (2011) Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. *Information and Software Technology, 53*(7), 693-706.

LEVINA N and VAAST E (2008) Innovating or doing as told? Status differences and overlapping boundaries in offshore collaboration. *MIS Quarterly, 32*(2), 307-332.

MAHNKE V, WAREHAM J and BJORN-ANDERSEN N (2008) Offshore middlemen: Transnational intermediation in technology sourcing. *Journal of Information Technology, 23*(1), 18-30.

MATHEW SK (2011) Mitigation of risks due to service provider behavior in offshore software development. *Strategic Outsourcing: An International Journal, 4*(2), 179-200.

MIRANI R (2007) Procedural coordination and offshored software tasks: Lessons from two case studies. *Information & Management, 44*(2), 216-230.

NAKATSU RT and IACOVOU CL (2009) A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel delphi study. *Information & Management, 46*(1), 57-68.

NATOVICH J (2003) Vendor related risks in IT development: A chronology of an outsourced project failure. *Technology Analysis & Strategic Management, 15*(4), 409-419.

NICHOLSON B and SAHAY S (2001) Some political and cultural issues in the globalisation of software development: Case experience from britain and India. *Information and Organization, 11*(1), 25-43.

NURMI A, HALLIKAINEN P and ROSSI M (2005) Coordination of outsourced information system development in multiple customer environment - a case study of a joint information system development project. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pp. 260a, Hawaii, HI.

OZA N, HALL T, RAINER A and GREY S (2004) Critical factors in software outsourcing: A pilot study. In *Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research* (MEHANDJIEV N and BRERETON P, Eds), pp. 67-71, Newport Beach, CA.

PALVIA S (2008) Challenges for small enterprises in the sourcing life cycle: Evidence from offshoring to India. *Journal of Information Technology Case and Application Research, 10*(4), 75-84.

PRIES-HEJE J, BASKERVILLE R and HANSEN GI (2005) Strategy models for enabling offshore outsourcing: Russian short-cycle-time software development. *Information Technology for Development, 11*(1), 5-30.

RAI A, MARUPING LM and VENKATESH V (2009) Offshore information systems project success: The role of social embeddedness and cultural characteristics. *MIS Quarterly, 33*(3), 617-A617.

ROY V and AUBERT BA (2002) A resource-based analysis of IT sourcing. *ACM SIGMIS Database, 33*(2), 29-40.

SABHERWAL R (1999) The role of trust in outsourced IS development projects. *Communications of the ACM, 42*(2), 80-86.

SABHERWAL R (2003) The evolution of coordination in outsourced software development projects: A comparison of client and vendor perspectives. *Information and Organization, 13*(3), 153-202.

TIWANA A (2004) Beyond the black box: Knowledge overlaps in software outsourcing. *IEEE Software, 21*(5), 51-58.

TIWANA A (2010) Systems development ambidexterity: Explaining the complementary and substitutive roles of formal and informal controls. *Journal of Management Information Systems, 27*(2), 87-126.

VERNER JM and ABDULLAH LM (2012) Exploratory case study research: Outsourced project failure. *Information and Software Technology, 54*(8), 866-886.

WAREHAM J, MAHNKE V, PETERS S and BJORN-ANDERSEN N (2007) Communication metaphors-in-use: Technical communication and offshore systems development. *IEEE Transactions on Professional Communication, 50*(2), 93-108.

WINKLER JK, DIBBERN J and HEINZL A (2008) The impact of cultural differences in offshore outsourcing - case study results from German-Indian application development projects. *Information Systems Frontiers, 10*(2), 243-258.

ZATOLYUK S and ALLGOOD B (2004) Evaluating a country for offshore outsourcing: Software development providers in the ukraine. *Information Systems Management, 21*(3), 28-33.

Chapter 3: Essay 2 – Experiencing Agility in ISD Projects

Abstract

Agile information systems development (ISD) methods have enjoyed increasing popularity since the publication of the agile manifesto. While extant research has highlighted critical challenges and key benefits associated with the adoption of those methods, little is known with regards to the *experience of agility* in ISD projects. This issue motivated the present study, which addresses the question of “how do project teams experience agility in ISD projects?". To answer this question, we built on extant literature and extracted four main facets of agility in ISD: (1) cooperation, (2) flexibility, (3) learning and (4) leanness. We adopted a multiple case study design and collected data in five agile ISD projects. Our analysis revealed tensions, a concept found in organization theory and defined as inconsistencies between elements that are otherwise sensible when taken individually, as important elements of agile ISD projects. Building on these findings we propose (1) Agile::Org tensions between principles of agility in ISD and the organization where the project takes place and (2) Agile::Agile tensions among principles of agility in ISD. We argue that Agile::Org tensions are best addressed by social mechanisms while Agile::Agile tensions are best addressed by technical mechanisms that enhance the team’s velocity. Finally, we posit that tensions can be interdependent when a mechanism fails to successfully address a tension, rendering another tension salient. Accounting for the complex and uncertain nature of agile ISD projects, we make a theoretical contribution to the literature on agile ISD by building an explanation of the role and impacts of tensions on a team’s experience with agility in ISD.

Keywords

Information systems development, agile ISD, agile software development, agility, tensions, multiple case study, qualitative research.

3.1. Introduction

Agile information systems development (ISD) methods are one of the most popular means to conduct ISD projects (VersionOne.com, 2013). These methods find their roots in the agile manifesto, an online document published by a group of software engineers in 2001 (Beck et al., 2001). The authors of the manifesto proposed four core values and twelve principles advocating for *agility* in the information systems development (ISD) process. Since the publication of the manifesto, a number of agile ISD methods have emerged, such as Scrum and eXtreme Programming. Agile ISD methods emphasize the need to empower individuals, perform work in short development cycles, and involve customers throughout the entire development process. Notwithstanding some of the benefits associated with the use of agile ISD methods (Lee & Xia, 2010), it has been observed that the adoption of those methods often remains problematic (Conboy et al., 2011).

Our review of empirical works on the topic highlights an implicit assumption and a gap motivating our research. First, research on agile ISD reflects an implicit assumption equating the successful adoption of agile ISD methods with agility in ISD. We suggest that studying agility in ISD without focusing exclusively on methods may help us gain a better understanding of the challenges faced by teams in ISD projects as well the mechanisms that can help to address those challenges. Second, in line with the majority of practices prescribed by agile ISD methods, the literature has focused on interactions among actors involved in agile ISD projects. While this has greatly helped us gain a better understanding of the relevance of mechanisms such as control (e.g., Maruping et al., 2009) and coordination (e.g., Strode et al., 2012) in agile ISD projects, it fails to take into account the role that technology can play to support agility in ISD. We suggest that research accounting for mechanisms relying on technology in addition to those relying on interactions among actors may help us gain a more comprehensive understanding as to how teams experience agility in ISD projects.

Given these gaps in extant research, and seeking to better understand agility in ISD, this work addresses the overarching question of: “how do teams experience agility in ISD projects?” Specifically, we ask: (1) what are the challenges teams face in agile ISD projects; (2) what are the mechanisms contributing to address those challenges; and (3) what are the impacts of those mechanisms on projects. To answer our research questions, we used a multiple case study design (Eisenhardt, 1989). We collected qualitative data in the form of semi-structured interviews and field observation from five agile ISD projects in three organizations. Alternating between our chains of evidence and our raw data through multiple rounds of coding, we found that some elements that appeared sensible when taken individually were inconsistent when considered together. We thus considered these *tensions* between elements of ISD projects as important challenges and we analyzed the mechanisms that served to address those tensions and their effect.

Building on this analysis, we propose a *tension-based* model of how teams experience agility in ISD projects, where tensions are defined as “elements that seem logical individually but inconsistent and even absurd when juxtaposed” (Smith & Lewis, 2011:382). Answering our first research question, our model suggests that within the context of agile ISD projects, two main categories of tensions exist: (1) Agile::Org tensions, referring to inconsistencies between the principles of agility in ISD (e.g., having users fully committed to the project) and the organization (e.g., having users remain committed to their main tasks outside of the project); and (2) Agile::Agile tensions, referring to inconsistencies among principles of agility in ISD (e.g., delivering a software artifact with speed while ensuring system quality).

Answering our second research question, our model suggests that social and technical mechanisms contribute to address tensions. Social mechanisms, defined as activities that rely on interactions between actors involved in the project, aim at: (1) spanning boundaries among team members’ specialized skillsets and between the team and the rest of the organization; (2) controlling the variability of elements within the project to ensure that those elements can vary based on unforeseen contingencies (e.g., changes in requirements) while ensuring that amplitude of those variations remains within acceptable limits; and (3) monitoring failure to help the team use failure as an opportunity for change rather than an undesirable event. Technical mechanisms, defined as activities that rely on the use of technology, aim at: (1) supporting value-adding activities through the availability of efficient and effective technological platforms (e.g., integrated collaboration technologies); and (2) automating non-value adding tasks to enhance the team’s velocity.

Answering our third research question, we suggest that, on the one hand, social mechanisms are best suited to address Agile::Org tensions because those tensions rely primarily on interactions among actors. On the other hand, we suggest that technical mechanisms are best suited to address Agile::Agile tensions because they can provide the degree of support or automation required to increase the speed of delivery while ensuring adequate levels of quality and communication. Finally, we suggest that social mechanisms may render latent Agile::Org tensions salient as they modify the patterns of interactions among actors involved in agile ISD projects.

Our work makes four contributions to the literature on agile ISD. First, we extend current research by explaining the role of tensions in ISD projects, arguing that these tensions may be constituted by inconsistencies (1) between elements of the principles of agility in ISD and the organization (Agile::Org tensions); or (2) among principles of agility in ISD (Agile::Agile tensions). Second, we highlight the instrumental role that technical mechanisms can play in the context of Agile::Agile tensions. Third, we shed light on the interdependencies that may exist among tensions and the possibility that social mechanisms can render latent Agile::Org tensions salient. Finally, we develop a specification of agility in ISD based on four overarching facets that may serve as a basis for refining the conceptualization and operationalization of the construct of agility in ISD.

Our work also makes a methodological contribution to qualitative research. While works on qualitative research methods provide ample descriptions of the various steps involved in the coding and analysis of qualitative data, the creation and maintenance of chains of evidence, one of the most prominent tools used during data analysis, is largely a manual process. The techniques developed in this work have helped us organize and analyze chains of evidence. Specifically, the use of a relational engine afforded a number of facilities that helped perform within case analysis and scale our analysis across cases. Overall, the techniques we have developed for this work have helped us greatly with the discovery of the emergent patterns that are the foundation of theory-building from case study data (Eisenhardt, 1989).

In the next sections, we review the literature and highlight the main gaps motivating our work before describing the methods we used for our empirical investigation. We then present our results followed by our theoretical explanation. Finally, we outline the main contributions and limitations of our work.

3.2. Literature on Agile ISD

The origins of agile ISD can be traced back to the publication of the Agile Manifesto (Beck et al., 2001), an online document signed by a group of software engineers. The manifesto advocates for the inclusion of agility within the ISD process through four core values and twelve principles (see Table 3.1). The manifesto favors an approach where (1) customers are deeply involved in the development process, (2) teams are autonomous and most importantly, (3) software artifacts are developed incrementally in short iterations to quickly deliver value for customers. It has been argued that the iterative and incremental approach to ISD advocated by the manifesto is not new (Conboy, 2009; Erickson et al., 2005; Larman & Basili, 2003). However, the notion of agility in ISD explicitly emerges with the publication of the manifesto and prior to this, with the publication of only two self-reported case studies (Aoyama, 1998a, 1998b).

Table 3.1. Agile manifesto values and principles
(Beck et al., 2001)

VALUES
Individuals and interactions over processes and tools **Working software** over comprehensive documentation **Customer collaboration** over contract negotiation **Responding to change** over following a plan That is, while there is value in the items on the right, we value the items on the left more. [emphasis original]
PRINCIPLES
• Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. • Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. • Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. • Business people and developers must work together daily throughout the project. • Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. • The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. • Working software is the primary measure of progress. • Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. • Continuous attention to technical excellence and good design enhances agility. • Simplicity--the art of maximizing the amount of work not done--is essential.

- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

We searched the Web of Science to extract peer-reviewed articles and conference proceedings published between 2001 and 2015 using the following search string: TOPIC: ((“agile” OR “agility”) AND (“software” OR “systems”)). This process yielded 1,812 results. An initial screening was performed to remove articles on the topic of agility outside of the context of ISD (e.g., in supply chain management) and 285 works were retained. We then restricted this result set to works containing first-hand empirical evidence by searching for abstracts, which led to a final list of 86 works, all related to the use of agile ISD methods. The resulting works were then individually read and coded against our three research questions. The analysis of those works resulted in three main observations motivating our research.

Agility in ISD. Given our focus on agility – instead than on agile methods – we first searched for a clear, agreed upon, and usable conceptual definition of the term. We noted, in line with empirical works (Sarker et al., 2009), conceptual works (Conboy, 2009) as well as literature reviews on agile ISD (Dybå, 2002; Hummel, 2014) that the concept of agility in ISD still lacks clarity. Research on agile ISD has matured greatly over the years (Dingsøyr et al., 2012) as it migrated from a practitioner-oriented literature (e.g., Agile Conferences) to a legitimate stream of research. Yet works typically approach the concept of agility in ISD based on the features of a specific method (Hummel, 2014). Within the IS discipline, this lack of clarity has been singled out as one of the main hinderers of the building of a cumulative knowledge base on the topic (Conboy, 2009).

While the study of agile ISD methods and their benefits continue to be the subject of many works (Balijepally et al., 2009; Lee & Xia, 2010), others have offered definitions of agility in ISD based on extant definitions found in other fields (e.g., Sarker & Sarker, 2009; Vidgen & Wang, 2009). We inventoried the most commonly used research-based definitions of this concept based on the results of the systematic review performed by Hummel (2014). We present those definitions in Table 3.2.

Table 3.2. Extant definitions of agility in ISD

Source	Definition	Foundation
Abrahamsson et al. (2002:17)	"What makes a development method an agile one? This is the case when software development is incremental (small software releases, with rapid cycles), cooperative (customer and developers working constantly together with close communication), straightforward (the method itself is easy to learn and to modify, well documented), and adaptive (able to make last moment changes)"	Existing agile ISD methods
Boehm and Turner (2004:718)	"Agility applies memory and history to adjust to new environments, react and adapt, take advantage of unexpected opportunities, and update the experience base for the future"	Unspecified
Qumer and Henderson-Sellers (2006:505)	"Agility is a persistent behavior or ability of a sensitive entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, uses economical, simple and quality instruments in a dynamic environment and applies updated prior knowledge and experience to learn from the internal and external environment"	Unspecified
Conboy (2009:340)	"The continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment"	Other disciplines and previous work on the topic (Conboy & Fitzgerald, 2004)

A recurring argument found in works where those definitions are provided as well as several others (e.g., Rodriguez et al., 2012; Sarker et al., 2009) is that agility in ISD is a complex concept that may be best defined through its constituting facets. In line with this view, we proceeded to decompose the definitions presented in Table 3.2 and also used the deductive specification of the dimensions of agility in ISD developed by Wufka (2013). We grouped those dimensions to identify the main constituting facets of agility in ISD, as illustrated in Table 3.3. To describe those facets, we went back to the roots of the agile ISD movement, the twelve principles of the Agile Manifesto. In line with the definition of a principle as "a moral rule or belief that helps one know what is right and wrong and that influences one's actions" (Merriam-Webster Online, 2015), the signatories of the manifesto state that they *follow* its twelve principles. This anchors ISD as a social process (e.g., Cockburn & Highsmith, 2001; Fitzgerald et al., 2002) that is best studied in an *amethodical* (Truex et al., 2000) manner, that is, outside of the boundaries of a given ISD method. We suggest, based on past literature on the topic, that agility in ISD comprises four facets: cooperation, flexibility, learning and leanness anchored in the principles of the agile manifesto.

Table 3.3. Overarching Facets of agility in ISD

Facet	Corresponding dimensions found in other definitions
Cooperation: ability for a group of individuals involved in an ISD project to work together	• Cooperative (customer and developers working constantly together with close communication) (Abrahamsson et al., 2002)
Flexibility: ability for a group of individuals involved in an ISD project to sense the need for change and respond to it promptly	• Adaptive (able to make last moment changes) (Abrahamsson et al., 2002) • React and adapt (Boehm & Turner, 2004) • Take advantage of unexpected opportunities (Boehm & Turner, 2004) • Flexibility (Conboy, 2009) • Responsiveness (Qumer & Henderson-Sellers, 2006) • Flexibility (Qumer & Henderson-Sellers, 2006) • Early recognition of the need for changes (Wufka, 2013) • Quick response to recognized required changed (Wufka, 2013)
Learning: ability for a group of individuals involved in an ISD project to build on past experience to adjust their internal processes	• Agility applies memory and history to adjust to new environments (Boehm & Turner, 2004) • Update the experience base for the future (Boehm & Turner, 2004) • Learning (Qumer & Henderson-Sellers, 2006) • Process agility (Wufka, 2013)
Leanness: ability for a group of individuals involved in an ISD project to rapidly produce software using principles of economy, simplicity and quality	• Straightforward (the method itself is easy to learn and to modify, well documented) • Incremental (small software releases, with rapid cycles) (Abrahamsson et al., 2002) • Leanness (Conboy, 2009) • Leanness (Qumer & Henderson-Sellers, 2006) • Speed (Qumer & Henderson-Sellers, 2006) • High degree of tangibility of intermediate results (Wufka, 2013) • Low overhead/leanness (Wufka, 2013)

Agility as the adoption and use of agile methods. Our review of the literature led us to observe that agile ISD research makes an implicit assumption equating the adoption and use of agile ISD methods with the experience of agility in ISD. Answering the concerns of practitioners, research has studied how agile methods such as eXtreme Programming – XP – (Beck & Andres, 2004) or Scrum (Schwaber & Beedle, 2003) are adopted in organizations. This literature shows that the practices prescribed by those methods allow teams to better answer changing requirements (Lee & Xia, 2010) by simplifying interactions among the actors involved in a project. For example, collocation and user participation in the development process allow team members to better understand user requirements and elicit rapid feedback loops (Helquiset et al., 2011).

While sensible, this approach implies that agility cannot be experienced in contexts where agile methods are not used. In addition, it implies that agility can be reached with the mere application of a method. These assumptions raise the question of the ability for a team to experience agility within a context where a method is customized or altogether absent. Indeed, method tailoring and customization represent large bodies of research in ISD and it has been argued that in most cases, there is a significant difference between the formal definitions of methods and their use as “methods in action” Fitzgerald et al. (2002:13). Our analysis of the literature supports this observation, revealing that not all agile ISD practices are deemed useful in some cases (e.g., Bowers et al., 2007) and that in other cases, practices are customized to better fit the organization where they are implemented (e.g., Fitzgerald et al., 2006).

The other issue with this assumption is that it pins agile ISD method adoption as an overarching objective that is enabled or hindered by a number of factors. For instance, a number of works provide lessons learnt (Turner & Boehm, 2003) and challenges (Boehm & Turner, 2005) regarding the adoption of agile ISD methods without considering the role of agility as a means to execute an ISD project. We need to better understand how teams experience agility in ISD projects as they take advantage – or not – of factors inhibiting or enabling the adoption of agile ISD methods as well as other, unexplored factors that may come into play within the context of a specific project. One work partially reflecting this perspective is that of Vidgen and Wang (2009) where the authors studied two ISD projects from the perspective of complex adaptive systems (CAS). They uncover enablers and inhibitors of agility in ISD based on three principles of CAS: (1) matching coevolutionary change rate; (2) optimizing self-organization; and (3) synchronizing exploitation and exploration. Their work highlights the need to better understand how challenges raised by those factors are addressed by teams during the course of a project.

A focus on social mechanisms. We also observed the predominance of interactions among actors as the main element supporting agility in ISD. Consistent with the vast majority of prescriptions found in agile ISD methods and in the principles of the agile manifesto, the literature has focused on the study of processes that revolve around those interactions. For instance, some of the most frequently used theoretical perspectives in the literature study control mechanisms (Maruping et al., 2009), coordination practices (Strode et al., 2012) and communication (Pikkarainen et al., 2008). This observation is in line with the preoccupations of practitioners who have sought to use agile ISD methods outside of their original boundaries. For example, several works study the customization of agile methods in the context of distributed ISD (Ramesh et al., 2012) or offshore ISD projects (Sarker & Sarker, 2009) and the authors of Scrum and XP, in the new editions of their books, provide some general guidelines to scale their methods in such contexts.

Notwithstanding, we find that the focus on interactions among actors forgoes the important role technology can play to support agility in ISD projects. For instance, the development of cloud computing platforms and other techniques such as test-driven development (Beck & Andres, 2004) may help teams achieve leanness by allowing for greater speed of delivery. The authors of XP mention the "ten minute build" as one of XP's *primary practices.* However, those technology-driven practices are rarely discussed in the literature. In the software engineering and computer science literatures, they are approached from the perspective of optimizing them rather than using them as a support for agility. We argue that including the role of technology along that of interactions among actors to understand how challenges are addressed in agile ISD projects can help further our understanding of the experience of agility in ISD projects.

Seeking to address those gaps and contribute to the literature on agile ISD in a rigorous manner, our objective is to offer a theoretical explanation as to how teams experience agility in ISD projects. In the next section, we provide the details of the methods we used to address our research questions.

3.3. Research Methods

Given our theory building objective, we opted for a multiple case study design and followed Eisenhardt's (1989) principles on the topic (p. 533), starting with an a priori specification of our focal construct, agility in ISD. We also referred to recommendations offered by Sarker et al. (2013) to guide the level of detail reported in the description of our research methods.

It has been argued that the multifaceted nature of the concept of agility in ISD renders the building of an overarching definition difficult (Lee & Xia, 2010; Rodriguez et al., 2012; Sarker et al., 2009). In line with this view, we used the four facets – cooperation, flexibility, learning, and leanness – extracted from our review of the literature on the topic as our a priori specification of agility in ISD.

3.3.1. Unit of analysis and case selection

Our unit of analysis is the ISD project. While our main research question relates to teams, it is within the context of ISD projects that agility in ISD is experienced by those teams. This accounts for the possibility that a given team may experience agility differently in two different projects because each of those projects has its own context and presents different challenges.

To reinforce the generalizability of our findings, our case selection strategy was based on the objectives of literal and theoretical replication (Yin, 2013). Literal replication involves the selection of cases to encourage the emergence of similarities across cases during data analysis. Theoretical replication involves the selection of cases to encourage the emergence of differences across cases during data analysis. To achieve this objective, we selected cases: (1) within the same organization as well as in different organizations; (2) where agility was perceived as important by project stakeholders even if no agile ISD method was actually used; (3) where different types of artifacts were developed; and (4) where artifacts were developed for internal as well as external customers.

3.3.2. Entry in the Field

Initial contacts were made with two firms in April 2013. The first one, Logistics, is a small North American software development firm. Contact was made possible because the author worked at Logistics between 2003 and 2011. The second one, AgileFirm, is a small North American consulting firm specializing in the development of business intelligence systems using agile ISD methods (Scrum). Contact with AgileFirm was made through personal acquaintances. Seven of AgileFirm's clients were initially contacted. Two of those clients showed interest in our research and granted access to their agile ISD project portfolio. Following these initial talks and the signing of confidentiality agreements, we started data collection for our first case in January 2014 and finished collecting data for our last case in September 2015.

3.3.3. Presentation of cases

Our sample included five cases – four ongoing projects and one retrospective project – in three organizations. The first organization, Entertain, is a large North American (10,000+ employees worldwide) firm specialized in the entertainment industry. Given its large size, Entertain has multiple ISD project ongoing at any given point in time. Members of Entertain’s project management office first granted us access to one case. Following further discussions and after reviewing their project portfolio, we were granted access to two other cases. The second organization, Insurance, is a North American subsidiary of a large European insurance company. The third organization, Logistics, is a small North American software development firm that sells a transactional system using a software-as-a-service model. Table 3.4 provides an overview of each case along with their respective data sources.

Table 3.4. Presentation of cases and data sources

Case	MarketBI	EventPlan	ResourceMgmt	ScheduleMgmt	LegalBI
Organization	Entertain	Entertain	Entertain	Logistics	Insurance
Type of IS	Marketing intelligence	Operational application	Operational application	Operational application	Business intelligence
Purpose	Assist decision making	Schedule events	Manage resources	Schedule resources	Report for regulatory compliance
Customer	Internal	Internal	Internal	External	Internal
Technology	Web application	Web application (cloud-based)	Web application (cloud-based)	Web application (cloud-based)	Back office application
Official ISD method	Scrum	Scrum	Scrum	None	None
Core team size	8	7	9	8	6
Data sources	Interviews (8) Field observation (7 hours)	Interviews (5) Field observation (7 hours)	Interviews (3) Field observation (11 hours)	Interviews (8) Field observation (5 hours)	Interviews (5)*
Respondents	•Client •Director of practice •Developers (2) •Functional analyst •Project manager •Solution architect •User representative	•Business analyst •Developers (2) •Project manager •User representative	•Developers (2) •Project manager	•Business analyst •Developers (4) •President •Software architect •Technical manager •VP product development	•Client •Delivery manager •Developer •QA Lead •Project manager
Data collection period	March 3, 2014 – June 27, 2014	April 17, 2014 – July 9, 2014	November 17, 2014 – September 14, 2015	May 26, 2014 – April 14, 2015	December 18, 2014 – March 12, 2015

(*) Data collection for LegalBI was performed retrospectively
Total number of pages of interview transcripts: 665
Total number of pages of field notes: 188

3.3.4. Instruments and Protocols

To better understand the context where teams experienced agility in ISD and triangulate our evidence, we relied on multiple sources of data. First, we conducted semi-structured interviews with various stakeholders involved in our cases (see Table 3.4). Interviews were performed face to face, recorded and transcribed after respondents agreed to participate and signed a confidentiality agreement. In two instances, interviews were not recorded due to respondents' preferences. Instead, detailed notes were taken during the interview and augmented as soon as the interview was finished. Additional questions were discussed with respondents via email or in person. The interview protocol and guide are included in Appendices 3.1 and 3.2.

Second, field observation was used to study the practices enacted by actors and complement interview data. In ResourceMgmt for instance, field observation occurred over the course of the entire project and allowed us to garner insights and perspectives from several actors that were not readily available given the small number of interviews we were able to conduct for that project. This additional source of evidence also provided important contextual information with regards to the physical location of team members, their personal relationships with one another, and the overall team dynamics. The author attended project events (e.g., planning meetings, software demos), gathering field notes by hand which were then typed and integrated in the case database. These notes included observation data as well as introspective data, along with other pieces of data such as non-verbal cues or room layouts. These data were supplemented by other forms of documentation (photographs, project documentation, correspondence between project members) when available.

3.3.5. Data analysis

Although we started our coding process using our specification of agility in ISD, we soon realized that the four facets – cooperation, flexibility, learning and leanness – were too broad to be used as our main codes. We relied on several techniques to iteratively analyze our data before integrating the four facets of agility in ISD in our analysis. First, we read all interview transcripts to identify high-level patterns. This process identified team members performing various actions in the context of ISD projects as an important element of analysis (e.g, "gaining access to users", "implementing workarounds").

Second, we selected a dedicated software (MaxQDA) to assist in the coding process. All data were imported in a case-based MaxQDA project.

We used inductive coding techniques borrowed from grounded theory (Glaser, 1978) and followed recommendations from Urquhart (2012) to perform within-case analysis. We performed (1) open coding to code our data based on content without the use of our conceptualization of agility in ISD. We then performed selective coding to group our open codes into "higher level categories" (Urquhart, 2012:193). Finally, we used theoretical coding to extract relationships between categories and relate those categories to the four facets of agility in ISD. Throughout these three stages, we used constant comparison (Urquhart, 2012:192) to ensure that the patterns emerging from our data were consistent within a given case as well as across cases. While we remained opened to the emergence of new facets, we were able to classify all our data within the four facets of our a priori specification of agility in ISD.

To assist in the selective and theoretical coding stages as well as the analysis of within-case and cross-case patterns, we built preliminary chains of evidence (Miles & Huberman, 1994) documented in a spreadsheet. However, the amount of data and the tabular format (i.e., cell-based) of the spreadsheet made it difficult to integrate all their constituting pieces to identify general patterns and discrepancies across cases. Therefore, we exported those chains of evidence into a relational database that we specifically designed for this work. Each case's preliminary chains of evidence were imported within an entity while a view provided a virtual entity linking all the cases' pieces of evidence together. A description of the logical model designed for this purpose is provided in Appendix 3.3.

A software artifact was also built to offer a user interface on top of the relational database to provide better visualization and to easily add or remove links between the preliminary chains of evidence and the coded data. Screenshots of this artifact are provided in Appendix 3.4.

To assist with data analysis, we also relied on the creation of queries against our data model to provide an alternative perspective on table-based, structured data contained within our chains of evidence. These queries reconstructed a sentence in plain English by taking pieces from a chain of evidence and linking them together, as illustrated in

Appendix 3.5. Combined with other, advanced queries, this process provided an alternative perspective on our evidence while summarizing findings from our data.

Analyzing patterns at the quote level provided detailed insights while grouping information allowed us to scale those detailed insights to a higher level understanding of our data. For example SQL's grouping, pivot and aggregation operators (GROUP BY, PIVOT, COUNT, SUM) can quickly perform complex operations and allow us to evaluate emergent theoretical insight against our data. Aggregation functions also allowed us to (1) find out in how many cases a chain of evidence was repeated; (2) discover cases where chains of evidence differed (e.g., given the same code, a different outcome occurs); and (3) validate that we were able to triangulate our evidence (e.g., counting how many quotes from different respondents were linked to a given chain of evidence).

Overall, this process provided a systematic approach to help structure and analyze our qualitative data. It is through this structured approach that we found some recurring patterns across cases that helped gain insight toward our research question. For example, we discovered that several actions performed by team members which were in line with prescriptions of agile ISD methods were associated with negative effects. Drilling down on those types of patterns led to the emergence of *tensions* as important challenges that affect how team experience agility in ISD projects.

We adopted the organization theory definition of tensions as "elements that seem logical individually but inconsistent and even absurd when juxtaposed" (Smith & Lewis, 2011:382). Each chain of evidence was re-evaluated to incorporate elements relevant to our three research questions. Chains of evidence were grouped together to create a matrix where we accounted for (1) the tensions teams faced in agile ISD projects, (2) the mechanisms that served to address those tensions, as well as (3) the impacts of those mechanisms on the project, answering each of our research questions. To determine whether a given tension was significant within a case, we studied (1) its occurrence in the interview transcripts and observation notes (e.g., as an important element that respondents recall from the overall project; as an element present in several observation sessions); and (2) the impacts associated with mechanisms contributing to address – or not – that tension.

We then grouped and labeled tensions across cases based on the facet of agility to which they belonged. To assess the link between a tension and one of the four facets of agility in ISD, we analyzed its constituting elements. For example, the “Variable scope versus Fixed scope” tension relates to *flexibility* because it deals with the ability for the team to adapt its workload based on what users and the rest of the organization dictate. Similarly, the “Speed versus Intrinsic quality” tension relates to the ability for the team to perform ISD by following principles of *leanness*.

3.4. Tensions in Agile ISD Projects, Tension-Addressing Mechanisms and Impacts

> “It’s clear that there’s a problem between agile management—in which I’m managing return on investment—, priorities and project management, which says, ‘No, no, no, we did what we said we’d do in terms of functionalities, don’t start pouring on gravy.’ ” [Business analyst, EventPlan]

It has been argued that tensions are an inherent part of organizational life and can lead to the emergence of paradoxical situations (Lewis, 2000; Quinn & Cameron, 1988; Van de Ven & Poole, 1988). Within this context tensions differ from other, related concepts such as dilemmas and dialectics. On the one hand, dialectics between two elements can be permanently resolved through their synthesis and dilemmas have clear pros and cons that guide the favoring of one element over the other. On the other hand, the constituting elements of tensions exist simultaneously and persist over time. While they can remain latent for long periods of time, tensions will become salient in one of two scenarios (Smith & Lewis, 2011). First, under conditions of plurality, change or scarcity that disturb trajectories that have been taken for granted within the organization. For example, in one of the firms we studied, Entertain, the project management office had previously declared that the IT department was taking an “agile turn” that had a profound impact on the way many IT projects would be managed.

Second, tensions can become salient under actors’ “paradoxical cognition” which makes those actors aware of tensions within their environment. For example, in project ScheduleMgmt, the VP of product development recognized that the project was trying to follow its roadmap while customers were demanding bug fixes that would have to be back ported into the current development version. In the next sections, we build on our data and present (1) categories of tensions in agile ISD projects and (2) the mechanisms that contribute to address those tensions as well as their impacts on the projects.

3.4.1. Categories of tensions in agile ISD projects

We identified two categories of tensions[1]. The first category – Agile::Org – is that of tensions between the principles of agility in ISD and the organization within which the agile ISD project takes place. An example of tension from this category is the ability for the team to be autonomous as advocated by the Agile Manifesto (Beck et al., 2001) and the need to consider the project as one of the many ISD initiatives within the organization.

The second category – Agile::Agile – includes tensions among elements of the principles of agility in ISD, such as the need for an ISD team to deliver quality and the need to deliver an artifact at frequent intervals, two principles found in the Agile Manifesto.

Our data suggest that the two categories of tensions we have identified pertain to the four facets of agility in ISD. Namely, Agile::Org tensions relate to cooperation, flexibility and learning and Agile::Agile tensions relate to leanness. Table 3.5 provides a list of the tensions identified within our data along with their most salient illustrative quotes and Table 3.6 provides a case-based inventory of the occurrence of those tensions.

To ensure a systematic presentation of our findings, the following pattern is used: (1) we first provide a short description and an illustrative quote of each tension; (2) we provide additional insight on the nature and relevance of this tension by incorporating elements from extant literature; (3) we present each quote by highlighting the tension it showcases based on its elements.

To present Agile::Org tensions, we underlined the agile element and put the organizational element in italics. To present Agile::Agile tensions, we underlined the first agile element and put the second agile element in a different font.

[1] Throughout the remainder of this text, we refer to these two categories of tensions as **Agile::Org** and **Agile::Agile** respectively. The '::' sign denotes the inconsistency of the elements of a tension and is the de facto convention in research on tensions and paradoxes (e.g., Smith & Lewis, 2011; Van de Ven & Poole, 1988)

Table 3.5. Overview of tensions

	Facet	Tension(s)	Illustrative quotes and observation notes
Agile::Org tensions	Cooperation	•Commitment (full commitment vs. *partial commitment* of external stakeholders)	“But being kept constantly informed of everything we’re doing…, *in fact, what my main client told me was that she wants to do even more in the next project*. I was like, ‘Yes.’ But the job of product owner is hands-on, it’s about someone being available. The resource has to be dedicated to the project…but really full-time. *If you aren’t there 100%, it just doesn’t work.*” [Business analyst, EventPlan]
		•Role (users as equals vs. *users as outside contributors*)	“So in the end we made her an integral part of the apparatus and she came to the scrum meetings […] *she had a tendency to, shall we say, warp things a bit.* […] She could, uh… reprioritize, which is not in itself a bad thing *but, you know, she had an opinion on everything.*” [Delivery manager, LegalBI]
		•Driving force (users as main drivers of the project vs. *organization as main driver of the project*)	“We drew up a 3-month scope with that, you know. And we never received authorization to spend money automating the payroll. You know, we ended up with an automated payroll and, at one point, with problems and delays, etc., but it was already done, because the link between developer and user is really so very close. […] *It’s just that, in terms of project management, it wasn’t part of the plan, it hadn’t been authorized.*” [Project manager, ResourceMgmt]
		•Resource skillset (multidisciplinarity across team members vs. *specialization*)	"*No-one handles more than one task, no-one is able to handle everything that’s going on here, from A to Z. Everyone can understand, but doing it, that’s a different story*. […] You’ve got your agile team and you say, in effect, usually you have little cards and everyone is able to take his card, and say, ‘OK, I’ll take this one, I’ll do this.’ When somebody else could have taken it. We hadn’t quite reached that point yet.” [Developer, MarketBI]
	Flexibility	•Scope (variable scope vs. *fixed scope*)	“*What was clear was that we were to replace the old system, to replicate all the current functionalities*. However, they wanted new functionalities, and we only heard about them as we went along. So, when we started the project, the backlog hadn’t been set, it hadn’t been configured. […] At the beginning, it was a bit like an open bar in terms of functionalities, and when we reached the end some things had to be cut, because there were things we had spent too much time on.” [Developer, ResourceMgmt]
		•Dependency (autonomy vs. *dependency*)	“*And then Architecture wanted us to set something up, and we did it,* even if, on the other hand…, even if within our team and for the manager, well, it didn’t make any sense.” [Developer, EventPlan]
		•Resource elasticity (fixed human resources vs. *elastic human resources*)	“*Cutting features wasn’t really a possibility. I think what happened is that we added resources. Quite a few*. Of course, as in anything, when you add resources, the potential gain (…), it’s just the opposite that happens, it’s always reduced a bit.” [Developer, ResourceMgmt]
	Learning	•Failure (opportunity for change vs. *undesirable event*)	“*At one point he’ll say, ‘Why don’t you have 40 days, why do you have only 30 days for this sprint?’* ‘There’s something that needs to be redone. Trust me, we’ll do it again.’ You hope that the conversation won’t go any further than that. It’s 10 days of catch-up on something that you have to take apart, but it was the best decision we could have made. *But if you had asked the client, that may not be what he would have told you.”* [Project manager, MarketBI] “I’ve seen some real sprint reviews, when we speak to each other directly and say: ‘Look, it bothers me what you’re doing’. *But in other circumstances, of course the scrum master will lead it, so that it’ll be positive, of course. So you doodle a bit and draw happy faces…*” [Business analyst, EventPlan]

	Facet	Tension(s)	Illustrative quotes and observation notes
Agile::Agile tensions	Leanness	•Pace and extrinsic quality (speed vs. *extrinsic quality*)	“ So, what I did, basically, was I looked at the figures to see if they were in the right ballpark to save some time *because in one week, forget it, testing enormous [integrated tests], you won’t be able to do it.*” [Developer, MarketBI]
		•Pace and intrinsic quality (speed vs. *intrinsic quality*)	“*I spent a sprint, a sprint and a half to tear down our database model again and then rebuild it.* […] you could certainly say that our speed was cut in half.” [Developer, ResourceMgmt]
		•Pace and communication (speed vs. *communication*)	“Holding meetings every 2 weeks for sprint planning, well it’s unmanageable, we just won’t make it. So that is why I went for 3 weeks and *I saw that, it’s true, 3 weeks is a lot. At the same time, there’s a cost to just… All the planning, all those things take a lot of meeting time, surprisingly.*” [QA Lead, LegalBI]

Table 3.6. Case-based overview of tensions

	Facet	Tension(s)	MarketBI	EventPlan	ResourceMgmt	ScheduleMgmt	LegalBI
Agile::Org tensions	Cooperation	•Commitment with external stakeholders (full commitment vs. *partial commitment* of external stakeholders)	▨	▨	▨		▨
		•Role (users as equals vs. *users as outside contributors*)	▨		▨		▨
		•Driving force (users as main drivers of the project vs. *organization as main driver of the project*)	▨	▨	▨	▨	
		•Resource skillset (multidisciplinarity across team members vs. *specialization*)	▨	▨	▨	▨	
	Flexibility	•Scope (variable scope vs. *fixed scope*)	▨	▨	▨	▨	
		•Dependency (autonomy vs. *dependency*)		▨	▨	▨	▨

		•Resource elasticity (fixed human resources vs. *elastic human resources*)			▨		▨
	Learning	•Failure (opportunity for change vs. *undesirable event*)	▨	▨	▨	▨	▨
Agile::Agile tensions	Leanness	•Pace and extrinsic quality (speed vs. *extrinsic quality*)	▨	▨	▨	▨	▨
		•Pace and intrinsic quality (speed vs. *intrinsic quality*)	▨	▨	▨	▨	▨
		•Pace and communication (speed vs. *communication*)	▨	▨	▨	▨	▨

Note : Cells in grey represent cases where a tension was present while cells in white represent cases where a tension was not present

3.4.1.1. Agile::Org Tensions – Cooperation

The Agile Manifesto and agile ISD methods such as Scrum, XP, or Crystal Methods place cooperation – the first facet of agility in ISD – as one of the most important aspects of the ISD process. To that end, agile ISD advocates close collaboration between developers and users (Beck et al., 2001; Schwaber & Beedle, 2003), increased user participation in the project, value creation as a main objective over traditional success factors – e.g., on time, on budget –, and the ability for team members to execute a wide array of tasks. Notwithstanding, our data suggest that cooperation is challenged by four main types of tensions that we present below.

Commitment with external stakeholders (full commitment versus ***partial commitment*** **of external stakeholders)**. Commitment tensions with external stakeholders pertain to the ability of external stakeholders such as user representatives to be fully dedicated to the project and *their ability to fulfill duties outside the project*.

> “ At one point I was told, ‘Oh well, often in projects in agile mode, as the client you need to be sitting in the team with us, working alongside us, like through every hour of the project, participating in scrums and doing it.’ *But given the time constraints, it was [someone else] who played that role for me.*” [User representative, EventPlan].

In all the cases where the artifact was developed for internal clients, the team expected user representatives to be committed to the project, while the rest of the organization expected them to continue fulfilling their daily tasks. As a result, the dedication of user representatives advocated by agile ISD was not feasible. For project teams, this meant suboptimal feedback loops, often resulting in delays, quality issues or extra work for developers who had to take on some of the tasks originally assigned to users.

> “The client also needed to come to our daily scrums, take part, make decisions and help us move things forward. So, is this what happened? Yes, in part it was. *Because, well, we shouldered a greater share of the burden, but he should have taken a bit more. But they’re so overworked because they’re cutting costs, too.*” [Developer, MarketBI].

Role (users as equals versus ***users as outside contributors*).** Role tensions pertain to the role of users as equal participants in the project and the perception from team members of *users as outside contributors*.

> “*Your role is really to give us the guideline on that, and the business analyst is supposed to turn up and say, ‘Yeah, I understand your guideline, except that the business is heading this way, so we’ll compromise. But this is how we’re gonna do it.’* […] And then at one point it has to stop. For certain things, we were really working hard but to no avail, precisely because the women dug in their heels and took power.” [Developer, ResourceMgmt]

While agile ISD advocates the participation of users and user representatives as equals from the rest of the team (Beck & Andres, 2004), teams sometimes perceived them as outside contributors whose role within the project was limited and did not allow them to voice their opinion on matters outside of those boundaries. This tension was especially relevant in ResourceMgmt where user representatives were involved in the project on a temporary basis. As a result, the rest of the ISD team viewed their role as limited within the project.

> "*There were so many things that they escalated that it was really out of scope, outside of their responsibility*. […] We found ourselves with an automated payroll and, at one point, with problems and delays and all that, but it was already a done deal, you know, because of the close working relationship between the developer and the user." [Project manager, ResourceMgmt]

In IS, the relevance of the link between an increase in user participation and positive outcomes in ISD projects has been studied outside of the context of agile ISD (Hartwick & Barki, 1994). Advocates of agile ISD take this relationship further by considering users as complete, full-time members of the team. This is based on the notion that users are best able to formulate their requirements, provide timely feedback and in some cases, contribute to tasks outside of their traditional responsibilities (e.g., drawing requirements), in line with calls for alternative conceptualizations of user participation in IS research (Markus & Mao, 2004).

In contrast, our data suggest that user representatives may create disruptions in the communication and coordination processes that support cooperation within the team. Again, ResourceMgmt provides a good illustration of this situation. While it was common for other team members (e.g., project manager, developers) to have meetings with ResourceMgmt's enterprise architecture team, it was never implied that user representatives could do the same.

> "They even had meetings with higher-level people, you know, people involved in the architecture and IT, to contest what IT was saying. *So, I mean, I'd say that their role…, their role was poorly defined at the outset and they took so much power, so that at the end it was extremely difficult.*" [Developer, ResourceMgmt]

Driving force (users as main drivers of the project versus ***organization as main driver of the project*)**. Driving force tensions pertain to the ability for users and user representatives to drive the development of an artifact and the *ability for other parts of the organization to drive the development of the artifact*.

> “*You [the organization] want to hit on me? Hit on me as much as you want*. My client is happy with the other end of the solution, so I’ve got an ally. *You want to hit on me for the scope, we agreed to sign the damn mandate at the start, so go ahead, we’ll talk about it*, I’ll bring my client with me and he’ll tell you that he’s happy and we were out in left field when we thought that we had to deliver a solution for Finance.” [Project manager, MarketBI]

In principle, agile ISD proposes users as the main drivers of development efforts. They determine, on a regular basis, which features should be developed over others and may decide to revisit requirements (e.g., Lee & Xia, 2010). At the same time, the organization may act as a driving force through scope and budget constraints as well as specific requirements (e.g., work within a given architectural framework), even though this functionality may appear devoid of value to the users.

Driving force tensions result from the need for teams to answer the requirements of different categories of stakeholders (e.g., users and the rest of the organization). Interestingly, this tension was also observed in ScheduleMgmt even though the product was developed for external clients. Our data suggest that this tension is especially relevant in this case because the development of functionality driven by external clients – i.e., users – was directly linked to financial incentives (payments upon delivery and penalties in case of delays) while the organization tried to follow a product roadmap. Team members struggled to reconcile these two elements that concurrently drove the development of the application, as illustrated by a developer:

> “*We had roadmaps nonetheless.* […] Of course if we see that there’s something we can do quickly to help the [users], we’ll do it.” [Developer, ScheduleMgmt]
>
> “I think it’s inevitable. *It’s hard to say, ‘OK, this version will be ready a month before the client goes live*.’ It’s the client who’s paying, and when there’s money involved, in the final analysis, that’s what drives a lot of what happens.” [Architect, ScheduleMgmt]

The only case where we did not observe this tension is LegalBI. Indeed, the requirements drawn for LegalBI emerged from the need for Insurance to comply with reporting rules and regulations. As a result, there were no particular instances where business stakeholders located within and outside the ISD team were at odds with the need to implement certain functionalities. Rather, all required functionality had to be implemented by the delivery date.

Resource skillset (multidisciplinarity across team members versus ***specialization*)**. The resource skillset tension pertains to the availability of multidisciplinary resources advocated by proponents of agile ISD and *the specialization of those resources based on the skills required to develop a software artifact*.

> "The fact that everyone is on equal footing and interchangeable, it's… It's absurd. *It's patently absurd because you can't ask a [user representative] to code, you can't…, I mean, even if you did, you won't have the same return, it just ain't gonna happen.*" [Developer, ResourceMgmt]

In line with arguments from other fields (e.g., manufacturing) where agility is relevant, agility in ISD is based on the idea that resources can be reconfigured with minimal disruption (Conboy, 2009). To enable this reconfiguration, members of an agile team must be able to conduct any of the development activities required at a given point in time. For example, developers in EventPlan were often in charge of conducting meetings. However, the levels of complexity of the requirements and the technology used in many software artifacts translated into an inability to have all members of an ISD team be able to perform any task based on current priorities. Indeed, having specialized resources who can take on a number of specific, unrelated tasks (e.g., server side development and requirements engineering) may not be feasible or very costly for the organization. As a result, the reality experienced by many projects is that true multidisciplinarity can only be obtained at the expense of specialization.

> "So, it's good to have a team that does everything, *but it's very difficult to achieve. It can be expensive, too.*" [Functional analyst, MarketBI]

While multidisciplinarity is viewed as an overarching principle of agility in ISD (e.g., Long & Starr, 2008), our data suggest that this tension was not only present in all but one of our cases but that it also had implications on the execution of the project. In principle, work should be assigned based on the priorities defined by users. In reality, work was assigned based on a *combination* of priorities defined by users as well as the availability of particular resources.

In LegalBI, this tension was not observed because all members of the team were senior consultants. This observation echoes arguments made by researchers who posit that the skills of team members are a critical factor in agile ISD projects (Erickson et al., 2005) although teams from our other cases presented a more diverse level of proficiency across team members, consistent with typical agile ISD projects (Balijepally et al., 2009).

> "We have a real swat team. We put people on it who are very strong in terms of business analysis, so they have great knowledge of the business. […] And we had senior people in terms of architecture, the design environment, and ETL, who had been on several other projects."[2] [Project manager, LegalBI]

[2] Since this quote reflects an absence of tension, no special formatting is applied.

3.4.1.2. Agile::Org Tensions – Flexibility

Recall that flexibility is one of the foundational principles of agility in ISD (Conboy, 2009) and the second facet of agility in ISD. Flexibility enables swift adaptation based on unforeseen contingencies. For example agile ISD advocates a reprioritization of work at regular intervals to ensure that the scope remains flexible. This is consistent with the idea of providing users with the ability to see the artifact “grow” (Brooks, 1995:14) over time. We find that in practice, three types of tensions pertain to flexibility in agile ISD.

Scope (variable scope versus ***fixed scope*)**. The scope tension pertains to the ability to consider the scope as variable and the tendency *for the organization to fix elements of the scope through constraints (e.g., budget)*.

> “*The governance framework imposes constraints on me in terms of budget approvals, for example. [...] I went out and got what I needed to finish what we had in the project’s scope*, but that generated frustrations, particularly for my business analyst. He believes – and rightly so – that this application is in a way his baby, or in some small way, that of other members on the team, and he’s the one who’s directly in contact with the client, the one with a good vision of what the client wants and would like to add to the system.” [Project manager, EventPlan]

In ResourceMgmt, the budget and scope were primarily determined by the need to replace a legacy application. Yet user representatives were also able to decide which functionalities should be implemented at the beginning of an iteration. In some instances, their priorities involved adding extra functionality that was not part of the allocated budget.

> “*Of course we identified all the major stories, meaning the large existing modules in the existing application, with all the use cases required in the application*. […] They pulled some fast ones on us, including a big payroll automation section *that was never in the scope*, but the attitude was, ‘As long as we’re doing the payroll, we might as well automate it, since we’re fed up with doing the payroll as [user representative].’ And you know, might as well this, might as well that… it added three months to the scope, you know. […] *And we were never authorized to automate the payroll.*” [Project manager, ResourceMgmt]

In EventPlan, this tension created frustration with users toward the end of the project because they felt let down by the team with regards to some of the functionality that could not be implemented. However, due to a lack of additional budget for the project, those new features were left for a future version.

> “Perhaps they [the users] would have wanted another iteration, perhaps two more, just to wrap things up well, to have a soft landing, but *I answered, saying, ‘I’m sorry, I have no more money to pay for another iteration and some of the improvements or fine-tuning you’d like to do*.’ That was the source of some frustration.” [Project manager, EventPlan]

Dependency (autonomy versus ***dependency*).** The dependency tension pertains to the ability for an ISD team to conduct the project autonomously and *the need to work with the dependencies of that project (e.g., other projects or teams)*.

> "First, we began developing integrations that we had previously discarded, and *that we had to begin all over again because the project took a new direction*. *The direction was discussed at great length, and it was put off until the end of January,* such that we wanted to begin, for example, developing integrations in September. *What we did wasn't any good, and we had to start all over again in January.* That put us under considerable pressure, in the end." [Project manager, ResourceMgmt]

In ResourceMgmt, team members were dependent on two other projects. However those dependencies worked on a different timescale. As a result, bug fixes and decisions as to how to implement certain functionalities were often received late, causing delays that eventually pushed back the release of the application.

> "We felt that we were, you could say, alone on the front lines. [...] *We also had some architecture problems in EventPlan, but it wasn't all that serious. It was really just that we waited, and waited, and waited. We didn't redo things.*" [Developer, ResourceMgmt]

In the definition of agile ISD methods (e.g., Beck & Andres, 2004; Schwaber & Beedle, 2003), the exemplars described by authors are independent projects where the team manages the whole development process up until the deployment of the application. However, our cases took place in contexts where organizations develop and integrate multiple IS, translating into an inability for a project to be truly autonomous.

Resource elasticity (fixed human resources versus ***elastic human resources*).** The resource elasticity tension pertains to the necessity to maintain a fixed team size and *the ability to expand or contract resources based on the current needs of the project*.

> "*[Project manager] also mentions that the new team members should bring some relief*. [Developer] warns her not to count them as full resources for a while. He tells her that she can count 10 hours per sprint for them but that anything they contribute should be a bonus because they will have a lot to catch up on before they are actually productive. [Developer] agrees and tells her not to be optimistic about them until they have proven themselves. [Project manager] reluctantly agrees." [Observation notes, ResourceMgmt iteration planning – 01/19/2015]

In agile ISD, small, collocated teams with a fixed number of members are preferred because they favor the development of the team spirit that enables better cooperation (Beck & Andres, 2004). In contrast, organizations typically view resources as elastic, preferring to expand or contract the size of a team based on the project's current needs. While favoring fixed over elastic resources may seem at odds with the objective of flexibility, our data suggest that adding extra members creates issues because there was a significant cost associated with bringing those new team members on board. Expanding resources required (1) transferring technical and business-related knowledge to new team members and (2) ensuring that new team members were able to follow the team's processes (e.g., daily meetings).

> "What you always need to keep in mind when starting a project is that *if one individual is brought on board, it has to be done very, very quickly*. And in fact there was such a case in the project, because at one point we had hired someone. So that made it a little… It was a period that was a little… yeah, I'd say it made for a bit of a rush." [QA Lead, LegalBI]

3.4.1.3. Agile::Org Tensions – Learning

To enable flexibility within the ISD process, agile ISD methods promote a number of activities pertaining to the third facet of agility, learning. For example, Scrum prescribes meetings at the end of every iteration to reflect on the efficiency and effectiveness of the ISD process and decide, in a consensual manner, on changes to that process. Consistent with elements of complex adaptive systems (CAS), learning enables *adaptation* to ensure fitness with the environment within which the agent – the team – is present (Holland, 1995). In agile ISD, Vidgen and Wang (2009) identified sharing and team learning as capabilities of agile teams enabling adaptation and coevolution. Notwithstanding the relevance of learning activities, our data reveals a tension between different perceptions of failure as they relate to learning.

Failure (opportunity for change versus ***undesirable event*)**. The failure tension pertains to the ability to use failure as an opportunity to reflect on the need for change and implement those changes and *the conceptualization of failure as an undesirable event*.

> "Our sprint reviews were so poor. Our sprint reviews, they were completely useless. Since in a real sprint review – and I've seen some real ones – we speak to each other directly and say, 'Look, it bothers me what you're doing. *But in other circumstances, of course the scrum master will lead the meeting to keep it positive. So you doodle a bit and draw happy faces…*" [Business analyst, EventPlan]

In ISD, teams often face the need to reflect on their work processes and reconfigure their resources as a result of changes in user requirements or technologies (Lyytinen & Rose, 2006; Mathiassen & Vainio, 2007). Agile ISD is based on the principle that "at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly" (Beck et al., 2001). Our data suggest that in some cases, teams did not perform learning activities to sense the need for change (LegalBI, ScheduleMgmt) while in others those activities were not followed by concrete actions to implement those changes.

> "Yeah, that's it, *we're learning despite ourselves. There was a lot of that, but all as a result of something that happened* rather than a vision." [VP Product development, ScheduleMgmt]
>
> "*It's enough to say, 'I had a problem' and it was, like, enough to explain the delay*. And that should be it in one or two cases, but eventually you have to do things differently. There was no pressure, either, when things didn't work out, so of course it didn't really make things any more tense." [Developer, ResourceMgmt]

3.4.1.4. Agile::Agile Tensions – Leanness

Leanness is an important facet of definitions of agility in ISD (e.g., Conboy, 2009; Qumer & Henderson-Sellers, 2006). Principles of economy, simplicity and quality are perceived as instrumental to enable the delivery of working software at frequent intervals while providing "technical excellence" and the "best architectures" (Beck et al., 2001). Within the agile ISD community, this has led to the emergence of Lean Software Development – LSD – (Poppendieck & Poppendieck, 2003). In agile ISD, leanness is enabled by social (e.g., collocation, self-organized teams) and technical mechanisms (e.g., continuous integration). Notwithstanding the benefits gained from the enactment of those mechanisms, our data suggest that their simultaneous achievement was often problematic and reveals three main types of tensions all relating to the notion of fast execution and delivery central to agile ISD[3].

Pace and extrinsic quality (speed versus *extrinsic quality***)**. Pace and extrinsic quality tensions pertain to the ability to deliver an artifact in short intervals and *the achievement of high levels of quality as perceived by users*.

> "And that, the internal people were experiencing it as a contradiction because, in the end, expectations had to be managed. I was the one doing that. In the sense of saying, 'Well, you want it quickly and *you want it debugged*. We needed to reach a compromise.' " [Project manager, LegalBI]

[3] Recall that in this section, the elements of a tension both emerge from principles of agility in ISD and prescriptions of agile ISD methods. When presenting tensions and quoting respondents, we underline the first element with a full line and *put the second in italics*.

While perceived system quality is an important variable in IS research (DeLone & McLean, 1992; Seddon, 1997) and agile ISD (Maruping et al., 2009), few works have studied the implications of attempting to *quickly* deliver high levels of quality (Baskerville et al., 2002). In MarketBI and LegalBI, the artifact under development was a business intelligence system that extracted data from heterogeneous sources, performed complex transformations on this data and loaded it into a data warehouse. Assessing the validity of this new data was difficult and time consuming for team members.

> "*The perception of the quality of our work* was closely linked to the person who found that either we weren't delivering files fast enough or that *they were of poor quality*. That happened quite a bit." [Delivery manager, LegalBI]

> "We hurried a bit and finally *we took a hit for it because it was missing a column. [...] Dammit, we hadn't finished testing, a step was missing* and then… We got a bit caught in a trap due to the client, who was asking for it and we hurried, and then *in the end we were criticized for it, we got a bit burnt.*" [Developer, LegalBI]

In other cases, the IS under development were transactional systems. As a result, the level of quality of the artifact could be assessed in several ways, including by user representatives interacting directly with the application. However, dependencies with other projects hindered the ability for the team to ensure quality while developing the artifact in short iterations.

> "We could have delivered in two months. […] Based on the business needs, it was a two to three-month project. With a lab framework, that's OK, we explore things, we test things, *we had some terrible regression effects. When we arrived with a new framework delivery at the same time as the version 2 delivery, a lot of screens stopped working, dropdowns that..., lots of things that just didn't work anymore.*" [Business analyst, EventPlan]

Pace and intrinsic quality (speed versus *intrinsic quality***)**. Pace and intrinsic quality tensions pertain to the ability to deliver an artifact at short intervals and *the achievement of high levels of quality as perceived by team members with regards to the internal structure of the artifact (e.g., architecture)*.

> " We needed to find a balance. What I mean is that [the solution architect] was so involved in the day-to-day work that *it was difficult for him to really have an overview of the solution* and that …, well, that resulted in us being broadsided by a couple of problems as the project moved forward, since things were changing so quickly." [Developer, MarketBI]

While principles of the Agile Manifesto advocate for the "best architectures" and "technical excellence", it has been argued that these objectives may be inconsistent with an ISD process that seeks to achieve a high velocity (Turner & Boehm, 2003). To achieve those objectives simultaneously, XP advocates refactoring, a practice aimed at modifying the internal structure of an artifact without affecting its external behavior (Opdyke, 1992). Refactoring in an important practice in software development (Mens & Tourwé, 2004; Murphy-Hill et al., 2012) and is promoted as a means to maintain technical excellence throughout the ISD process, thereby preventing the accumulation of technical debt.

> “ It’s that things were going fast, going fast, and *maybe a sprint will be added at the end for you to clean things up and arrange things, the odds and ends…that are left over. We’ll make some cuts, clean things up and it’ll be alright.*” [Director of practice, MarketBI]

In addition, our data suggest that some aspects of intrinsic quality were driven by dependencies rather than the team’s decision to ensure technical excellence. As a result, when those dependencies mandated changes to the artifact’s internal structure, the project’s velocity was hindered outside of the team’s control.

> “When we finally heard, ‘OK, here’s how it’s gonna work’ from Architecture, I did a sprint, a sprint and a half to tear down our database model again *and then rebuild it. Changing all the pages like that, without considering all the backups that are gonna work differently, but just the reading, how it will work, to rework it.* This is work with a rather limited ROI. […] Particularly since I doubt that it makes much of a contribution. There are costs to the user, in some cases.” [Developer, ResourceMgmt]

> “[The vendor] decided to drop its functionality in the latest version. And that required us to reposition very quickly, except that…, *that’s it, there are client projects that reflect the choices that were made at the time.*” [Head of architecture, Entertain]

Pace and communication (speed versus *communication***)**. The pace and communication tension pertains to the ability to deliver an artifact at short intervals and *the need to disseminate relevant information within the team as well as with the rest of the organization*.

> “ *It’s something that I need to use, and if everyone uses it, yes, we’ll benefit, and yes things will be clearer in terms of where we’re going and how long it will take to get there*. But I’d say that it doesn’t change anything if I’m in the process of doing something and, oops, five minutes later I have to create a new task because I’ll complete it in 30 minutes.” [Architect, ScheduleMgmt]

The Agile Manifesto and agile ISD methods advocate face-to-face communication as the primary means to convey information among team members. However, when interacting with the rest of the organization or with users who are not committed to the project on a full-time basis, frequent, short face-to-face conversations may not be feasible. In addition, practices prescribed by agile ISD methods to disseminate information across the team and in some instances (e.g., daily scrums) the rest of the organization can be time consuming. This tension is also echoed by some practitioners who consider that agile ISD methods prescribe too many meetings that bring little value for the team. For example, the first sentence of the Anti Agile Manifesto (Anti Agile, 2015) states that “We have suffered through countless consultants and hours of meetings”.

> “*The scrum* took too long. We sat down and there was just too much. It was the kind of scrum that could easily stretch into an hour.” [Director of practice, MarketBI]

3.4.2. Mechanisms Contributing to Address Tensions in Agile ISD Projects and their Impacts

We have presented tensions pertaining to the four facets of agility in ISD as important challenges in ISD projects. In this section, we analyze the mechanisms that contributed – or not – to address those tensions as well as their impacts.

In organization theory, tensions can spur vicious or virtuous cycles once they become salient (Smith & Lewis, 2011). Vicious cycles occur (1) when actors avoid the inconsistencies brought up by the tension (e.g., through denial), or when (2) actors reinforce one element of a tension over the other (e.g., by enforcing existing patterns of behavior). For example, in EventPlan, the project manager pretended to follow certain Scrum rituals (e.g., daily standup meetings) while using those meetings to reinforce the agenda of the organization, causing frustration within the team. Virtuous cycles occur when actors accept the tension and seek to take advantage of its elements by finding a way to accommodate them together. For example, in MarketBI, exploring architectural patterns helped the team tend to the intrinsic quality of the system early in the project without compromising the pace of delivery once development effectively started.

Following the identification of tensions through the analysis of our chains of evidence, we studied, within those chains of evidence and their related raw data, whether the issues associated with those tensions persisted over time. In some cases, we found that an action had taken place to address the tension at a later time. In other cases no action had taken place. These observations were then compared across cases to outline a series of *mechanisms* – activities that contribute to address a tension. We identified two types of mechanisms: (1) social mechanisms relying primarily on interactions between individuals and (2) technical mechanisms relying primarily on the use of technology and tools.

To evaluate the impacts of mechanisms on a given project, we performed a qualitative assessment to determine whether those mechanisms had an overall positive (e.g., an “improvement”, a “benefit”) or negative (e.g., a “loss”, a “waste”) effect on the project based on interview data as well as our observation notes. This procedure also applied to instances where no mechanism was present.

In line with conceptual (Smith & Lewis, 2011) as well as empirical (e.g., Jay, 2013) works on tensions and paradoxes in organizations, our data suggest that mechanisms had a positive impact on the project when they built on the two elements of a tension (e.g., achieving speed and extrinsic quality). When mechanisms emphasized one element over the other, those positive impacts were not observed and negative impacts were observed instead. In other instances, mechanisms were addressing a tension but had the side effect of raising other, related tensions.

The results of our analysis are presented in Table 3.7. For each facet of agility in ISD, Table 3.7 includes, for each type of tension identified in Table 3.6, the mechanisms that were found to contribute to addressing that tension. We also indicate instances where an impact was observed even when no mechanism was present. For example, the third row from the top in Table 3.7 shows that the mechanism “educating stakeholders” contributing to address role tensions had a positive impact in MarketBI and LegalBI. For each mechanism, we provide: (1) a short description and an illustrative quote; (2) additional insight on the mechanism as it relates to extant literature; (3) its impact on the project(s) where it was observed. Within the quotes, we have framed mechanisms with a single border, put their impacts in italics and framed underlying tensions revealed by those mechanisms with a double border. To contextualize the results of this analysis, we also include two illustrative vignettes that provide a longitudinal perspective on tensions and mechanisms. Table 3.8 showcases commitment and underlying role tensions in LegalBI and Table 3.9 highlights the negative impacts associated with the absence of mechanism addressing tensions in ScheduleMgmt.

Table 3.7. Overview of mechanisms to address tensions and their impact(s) on the project

	Facet	Tension	Mechanism(s)	MarketBI	EventPlan	ResourceMgmt	ScheduleMgmt	LegalBI
Agile::Org tensions	Cooperation	•Commitment (full commitment vs. *partial commitment* of external stakeholders)	•Using internal proxies		+			
			•Using external proxies	−		− Role		− Role
		•Role (users as equals vs. *users as outside contributors*)	•Educating stakeholders	+				+
			•No mechanism			−		
		•Driving force (users as main drivers of the project vs. *organization as main driver of the project*)	•Limiting transparency	+				
			•Enacting symbolic performances		−	−		
			•No mechanism				−	
		•Resource skillset (multidisciplinarity across team members vs. *specialization*)	•Pooling competencies	+	+	+		
			•No mechanism				−	
	Flexibility	•Scope (variable scope vs. *fixed scope*)	•Maintaining variability within budget constraints and limiting transparency	+				
			•No mechanism		−	−	−	
		•Dependency (autonomy vs. *dependency*)	•Rendering the impact of dependencies explicit		+	+		
			•Implementing technical workaround					+

			•Automating deployments				+	
		•Resource elasticity (fixed human resources vs. *elastic human resources*)	•Delegating independent tasks			+		+
	Learning	•Failure (opportunity for change vs. *undesirable event*)	•Monitoring failure internally and limiting transparency	+				
			•Symbolic performances		−	−		−
			•No mechanism				−	
Agile::Agile tensions	Leanness	•Pace and extrinsic quality (speed vs. *extrinsic quality*)	•Exclusively enrolling business	− Commitment		− Commitment		− Commitment
			•Automating tests		+		+	
		•Pace and intrinsic quality (speed vs. *intrinsic quality*)	•Limiting transparency	+				
			•Anticipating decisions		− Dependency	− Dependency		− Dependency
			•Automating deployments				+	
			•Exploring architectural patterns	+			+	+
		•Pace and communication (speed vs. *communication*)	•Performing short, frequent status update meetings and integrating collaboration technologies	+	+	+	+	

			•Using physical artifacts (board and post-it notes)			+		– Commitment

Notes:

Cells in grey represent cases where a tension was present.
Empty white cells represent cases where a tension was absent (see Table 3.5) and therefore no mechanisms were present.

Mechanisms describing social mechanisms are in plain text.
Mechanisms describing technical mechanisms are underlined.
Impacts are indicated as an overall positive (+) or negative (–) impact on the project.
Tensions raised by a mechanism are written inside double borders.

Table 3.8. An instance of a tension – mechanism – positive impact chain

In LegalBI, commitment tensions initially hindered cooperation because the team was unable to secure direct access to users. To address this tension, one of the employees at Insurance, a business analyst, was assigned as an external proxy. That person was supposed to act as a buffer between the team and the future users of the solution. However, some of LegalBI's team members felt that she hindered rather than eased up coordination with the rest of the organization by preventing them from having direct access to users, a situation which they felt would be more appropriate given the iterative nature of the project.

> "It would have been much more pleasant to be in direct contact with the customer. I have a question, I can talk to the customer, I see him, and we have a meeting. Here we experienced latency." [Delivery manager, LegalBI]

More importantly, as this person became involved in the team's work processes, she perceived her role very differently from what the team expected her to do, leading to the emergence of role tensions. For example, several respondents reported that she would physically move work items on the team's whiteboard or reprioritized work items in the middle of an iteration. In contrast, team members expected her to stick to acting as an interface with users and performing quality assurance work while being removed from the management of the team's daily tasks.

> "She would stand in front of the board and move post-it notes: 'Ok, this is a priority, I am putting in the iteration. This is done, I put in the done section'. But the developer, it is not done for him, he has not done his tests yet. […] There was some intrusion that was difficult to contain." [Developer, LegalBI]

As this tension became more prominent, the team decided to use education as a means to ensure that her role would remain within the boundaries that they expected her to abide to. For instance, the project manager gave her a crash course on agility and the execution of development cycles while other team members politely reminded her that she was not expected to prioritize work items in the middle of an iteration. This mechanism helped to address the role tension and following this episode, the team was able to retain the external proxy as an active contributor to the project who remained within her expected role for the project.

> "I went for coffee with her for an hour, an hour and a half. I told her about business intelligence and I gave her a sort of crash course at a very high level so that she would understand the concepts because she was neither familiar with agility nor was she familiar with business intelligence." [Project manager, LegalBI]

> "We told her: 'Wait, we manage the post-it notes on the whiteboard'. She understood very well." [Developer, LegalBI]

Table 3.9. An instance of a tension – absence of mechanism – increasing negative impact chain

In ScheduleMgmt, driving force tensions were problematic because the team was constantly pulled between the need to follow its internal product roadmap and the need to implement and debug a number of features for customers who had paid for those features. While this was identified by team members as well as the president of Logistics as an important issue, no mechanism was observed to address this tension. The president considered that this tension, while problematic, was temporary, unavoidable, and more importantly, perhaps unresolvable.

> "We made the decision that we had a big pill to swallow. […] We made the choice to live with it. We dealt with it. […] So that is the way we managed it, we said: 'It is going to hurt but you feel better after'." [President, ScheduleMgmt]

As the project went on, this tension was only exacerbated. First, preliminary versions of ScheduleMgmt were delivered to customers due to the need to fulfill contractual obligations. However, this created more issues because the code that was delivered was not fully functional. As a result, customers filed many bug reports and client issues and the team was expected to fix those issues while still working on the product roadmap. At that point, the team was working on (1) its internal product roadmap, (2) client features, (3) fixing bugs in a temporary version of its application to support customers who were running the untested version of the application, and (4) supporting new deployment procedures and migration of customers onto the new version of the application.

> “We had commitments to fulfill and we had no other choice, so we were pushed to release the code. A code that was not ready to go into production” [VP Product development, ScheduleMgmt]

> “When the time came to deploy the first customer, well it was not ready yet. In a sense that the code, the application was missing functionality. […] And for each existing client, there was a migration to perform. […] So I had to work during nights and my free time, either to think about or actually do things to deliver on time. During that time, I was stressed. […] I did not like that.” [Architect, ScheduleMgmt]

Eventually, this caused a great deal of stress within the team. Some team members thought of quitting while others were being pushed to deliver within very tight deadlines.

> “It was a morbid summer trying to keep active projects alive, to continue the development, trying to close down a version we were not able to finish, to continue to support sales. […] So I seriously considered quitting Logistics in August. […] It was not pleasant for the development team: everybody was depressed.” [VP Product development, ScheduleMgmt]

> “And they were brilliant, but you know, the [architect] who was so upset, it’s tough to see that happen. To think that I pushed the guy that far. And it comes down to us, it’s our fault: me, [the project portfolio manager], [the project manager] and [the president]. You know, it’s people who manage that, and we didn’t do our job that time. We really got ourselves in trouble at that point.” [VP product development, ScheduleMgmt]

3.4.2.1. Agile::Org Tensions – Cooperation

We grouped the mechanisms that contributed to addressing cooperation tensions under the term “spanning mechanisms”. Spanning aims at enabling the crossing of (1) structural and hierarchical boundaries with the rest of the organization (e.g., users, top management) and (2) skill-based boundaries within the team (e.g., backend and frontend developers). Below we present those mechanisms and their impacts.

Commitment (full commitment versus ***partial commitment*** **of external stakeholders)**. We observed two mechanisms contributing to address commitment tensions with external stakeholders (e.g., users). Both these mechanisms involve the use of key resources to counter the inability for users to be committed to the project on a full-time basis. In agile ISD research (e.g., Conboy & Fitzgerald, 2010; Mangalaraj et al., 2009; Strode, 2015) and in the words of our respondents, these *proxies* act as substitutes for users even though they are not part of traditional agile ISD methods.

> “The reality is that often the product owner… there’s someone in the team who’ll…, or someone in a client’s internal IT department who’ll play the role of the product owner’s proxy. […] In real life, it’s hard to have a real product owner who is as dedicated as we’d like in an agile project.” [Project manager, Consulting firm[4]]

The first mechanism, using internal proxies, is the use of team members as intermediaries between the team and external stakeholders.

[4] This quote is inserted to showcase the common use of this mechanism in agile ISD project and therefore does not include any special formatting.

> "[Business analyst] served a lot as an interface with [users]. […] He had analyzed all that. There had been some changes, that's only normal, but *the fact that he was able to feed us, […] it was good, though, because we could concentrate on that job.*" [Developer, EventPlan]

Internal proxies possess the required business and technical skills to translate user requirements into functional requirements. For instance, in EventPlan the business analyst acted as an internal proxy who was able to reconcile technical imperatives and user priorities, thereby facilitating coordination. In ScheduleMgmt, this mechanism had always been necessary since the artifact was sold to external customers who were never involved in the development process.

> "When clients…, I don't know their needs […] whereas the [VP product development] and the [President] have been there, […] *they've analyzed their processes and needs*." [Developer, ScheduleMgmt]

The second mechanism, using external proxies, is the use of resources sourced from external stakeholders and who act as intermediaries between the team and those external stakeholders.

> "And as I've said, the [user representatives] became the client, and *they started calling the shots, which put us in a real awkward position.*" [Project manager, ResourceMgmt]

In ResourceMgmt, using internal proxies triggered the emergence of role tensions. The distinction between internal and external proxies shows that in order to improve coordination with external stakeholders proxies must understand the team's work processes, and possess sufficient business and technical knowledge to interact with the team and the rest of the organization.

> "Normally the product owner, and even the proxy product owner, don't belong here, but we brought her in… Of course having access to our backlog, our … to our work and activities, *she could, well… reprioritize, which is not necessarily a bad thing, but you know she had an opinion on everything. Both on the methodology… sometimes on our methods…*" [Delivery manager, LegalBI]

Role (<u>users as equals</u> versus *users as outside contributors*). To address role tensions, we observed one mechanism. Educating stakeholders is a social mechanism that aims at ensuring that external stakeholders who participate in the project understand their role within the work processes defined by the team.

> "I said, […] '*If you want us to make some adjustments, you'll be able to adjust things along the way.*' […] And so we were educating at the same time. You know, when I said that things were slow at the start, well that was because we had to educate the client about agility." [Director of practice, MarketBI]

While educating stakeholders takes time, it is required to ensure that coordination between team members and those stakeholders is facilitated rather than hindered. Indeed, educating stakeholders is an important aspect related to the use of external proxies to address commitment tensions.

> "I went for a coffee with [the external proxy] – for about an hour, an hour and a half – and I spoke to her about business intelligence and gave her a very high-level crash course so that she'd have a somewhat better understanding of the concepts, since she didn't have much experience with agility, nor with business intelligence. So that reassured her and it…, *she felt more involved, so after that, it was a big help.*" [Project manager, LegalBI]

In ResourceMgmt, not educating external proxies proved problematic because the roles they fulfilled did not correspond to the roles that the rest of the team expected them to assume. For example, external proxies were reducing the priority of important backend functionality even though the rest of the team did not expect them to question the relevance of this work.

> "The [proxies'] role was poorly defined in the beginning and *they took so much power that, by the end, things got extremely difficult*. […] *As a result, we were really working hard but to no avail, precisely because the proxies pushed back and grabbed power.*" [Developer, ResourceMgmt]

Driving force (<u>users as main drivers of the project</u> versus ***organization as main driver of the project*****)**. We observed two mechanisms addressing driving force tensions. The first mechanism, limiting transparency, is a voluntary lack of transparency to communicate information to users and the organization.

> "I commit to deliverables at the beginning of a sprint and I give a status report on them at the end of the sprint. That's it, nothing more. Budget,… sorry, budget and deliverables, and that's it. […] *[Agility] allows you to make decisions without the client, without him being too aware of it, for his own benefit*. […] *So agility allows you to bypass these processes, because there's a kind of vagueness, and your job as project manager is to ensure that everyone is comfortable with this vagueness, comfortable enough to give you some flexibility.*" [Project manager, MarketBI]

Limiting transparency runs against the principles of agile ISD methods that advocate transparency as a means to disseminate information across all levels of the organization. Yet it allows the team to manage expectations from users and the organization by only communicating those pieces of information that are relevant to each group of stakeholders.

> "It's too complex to try to be 100% transparent." [Project manager, MarketBI]

The second mechanism, enacting symbolic performances, is the performance of agile rituals (e.g., daily scrums) that may give the impression that users are driving the development of the artifact while in reality, it is driven by the organization. While this mechanism was not discussed openly in ResourceMgmt, some of EventPlan's team members felt strongly about it.

> "I often raised the issue with [project manager]. I said, 'You're taking too big a role in our daily scrums because you're sending us project management messages on a daily basis. We couldn't care less about that stuff.' […] 'We did what we said we'd do in terms of functionalities, don't start pouring on gravy. He calls that gravy.' There's no issue of gravy here, we're here to improve the process. *So that's where there was a misunderstanding*." [Business analyst, EventPlan].

> "In fact, instead of saying, 'adopt agility,' I'd say that they [Entertain] want us to be more agile, with a small 'a', *without necessarily dropping their other, more traditional control objectives.*" [Project manager, EventPlan]

In ScheduleMgmt, no mechanisms were observed to address this tension, which led to some significant issues with regards to the Logistics' ability to follow its product roadmap and deliver a version on time.

> "Starting in January, our resources began to be *distracted from their work by clients who were already trying to use a version that was not entirely ready for use.*" [VP product development, ScheduleMgmt]

Resource skillset (multidisciplinarity across team members versus ***specialization*****).** To address resource skillset tensions, the same mechanism was observed across all cases. Pooling competencies is the building of a pool of core competencies so that at least part of the team is able to conduct a sufficient number of project activities.

> "I can analyze, I can do anything. […] *I presented myself as the project's developer and…, that's it. It's my pleasure to move the project forward.*" [Developer, EventPlan]

> "What you can do is have three business intelligence pools. You have a pool that is 100% front-end, another that is 100% back-end and a third deals with both. […] *So, it's good to have a team that does everything, but it's also very difficult. It can be expensive, too.*" [Functional analyst, MarketBI]

In agile ISD research, it has been argued that multiskilled individuals can help achieve multidisciplinarity as well as improve the reconfiguration of resources when needed (Sarker & Sarker, 2009). However, our data suggest that this mechanism is difficult to apply across an entire project. For example, MarketBI's project manager initially sought to achieve complete multidisciplinarity within the team. This strategy consumed a lot of time as it prevented team members from developing specific expertise. In all our cases, the technical and business-related knowledge required to develop the artifact made it impossible for all team members to perform all activities.

> “In a scrum team, you need people who are able to do just about everything. […] We don’t have the staffing budget for that, *so we organized around it as best we could*. This is a reality that [project manager] eventually came to appreciate.” [Developer, MarketBI]

In ScheduleMgmt, this mechanism was absent. As a result, there were very few resources capable of performing a variety of tasks (e.g., server-side and client-side development). While this helps team members become experts in their respective domains, it creates coordination issues when those resources are not available.

> “Well, [the architect] went on vacation, and *it became a real bottleneck because we were short on resources*. We had to call back, say, [the developer] on the project. He did what he could, but he asked [the architect] for review codes because he wasn’t comfortable with something, so that expertise… *You know, that led to some small problems.”* [VP product development, ScheduleMgmt]

3.4.2.2. Agile::Org Tensions – Flexibility

Mechanisms that contribute to addressing tensions that pertain to flexibility aim at controlling variability. Controlling variability encompasses activities that allow an ISD team to let some of the elements related to (1) scope, (2) dependencies, as well as (2) the elasticity of resources fluctuate while maintaining control over the amplitude of those variations.

Scope (variable scope versus ***fixed scope*)**. To address scope tensions, two combined mechanisms were observed in MarketBI. The first mechanism, maintaining variability within budget constraints, is the team’s commitment to budgetary constraints as a means to regulate variations in the scope requested by users.

> “At some level, *if I’m taking care of finances, then 80% of the answers will be provided* […]. *That’s all they want to know. [...] I make sure that my client is happy*. [...] I managed the budget on that basis. If I have so many weeks left, then I need to have so much budget left, in my back pocket. If I don’t, then I’ve got a problem. I’d even say that sometimes we spent the budget faster than planned. So what I did was, I reduced the scope of the sprint that followed.” [Project manager, MarketBI]

In plan-driven approaches, one of the measures of success is the delivery of all functionality included in the project’s scope. In agile ISD, schedule, costs, and scope constraints are also present but they are renegotiated at regular intervals as the project’s requirements evolve (Highsmith, 2010). Yet within those fluctuations budgets remain fixed. As a result, ensuring that the scope varies within the project’s budgetary constraints is more important than ensuring that all the elements of the original scope are included.

> “Yeah, [the budget] drives everything. […] People will ask you for an estimate, and you have no choice but to provide one. […] Is that alright? Is everyone good with this? Great. Once we have the budget, we break it all down. […] *So we’ve got our money, let’s go!*” [Project manager, MarketBI]

The second mechanism, limiting transparency, has been described previously. Limiting transparency allows the team to maintain a buffer zone that gives them some leeway with regards to the features that will be implemented.

> “I’ve never committed to an overall scope. Never, never, never. And in fact, I’m telling you that when I arrived, they had already committed to an overall scope. The first two steering meetings were to take it down completely and ensure that no one knew or got attached to it. […] *I arrive in front of my steering committee and say, ‘No, the overall scope has been changed.’ Yikes, that’s a big red flag, sirens, the whole works. It would take me two weeks to defuse all that and get my [product owner] together with the steering committee so that they… I don’t want that*. […] Trust me, it’s possible within our budget.” [Project manager, MarketBI]

In other cases, no mechanism was present. As a result, teams had difficulty ensuring that some of the elements that were mandated within the project were implemented. In those cases, the issue was not that the scope varied too often. Rather, it was that the scope fluctuated too much due to the addition of unplanned functionality that hindered the team’s ability to stay on budget and within some of the scope constraints that had been imposed upon them.

> “They challenged the architecture decisions, they called meetings with four enterprise architects. […] *They pulled several fast ones on us, including a big payroll automation module, which was never in the scope. […] And we never received authorization to spend money automating the payroll. You know, we ended up with an automated payroll and, at one point, with problems and delays and all the rest.*” [Project manager, ResourceMgmt]

Dependency (autonomy versus *dependency*). We have observed three main mechanisms addressing dependency tensions. The first mechanism, rendering the impact of dependencies explicit, is the presentation of clear evidence that allows the team to communicate the negative impacts of a dependency on the project.

> “I mean, you make architecture decisions that will have a big impact technically, and they don’t have our input on impact. *At one point they arrived and listened to us* – because we were bitching – […] We had it out […] until at one point they said, ‘OK, we won’t do it, we’ll put it off, it’ll be a technological debt.’ ” [Developer. ResourceMgmt]

Dependency tensions are difficult to manage because they involve stakeholders who may not be committed to the project on a full-time basis or be available when they are most needed. In addition, it may be difficult to predict when those actors will be needed during the course of an iteration. In EventPlan and ResourceMgmt, this created delays and quality issues because the teams were dependent on the schedules of other projects that were not aligned with their own. To resolve dependency issues, project managers had to escalate the impacts of those dependencies in a very explicit manner. However, this was only performed as a last resort.

> “It took so long that at one point the project manager wrote some sort of email saying, ‘Look, you [architects] aren’t able to answer our question, you’re going to make us late. I need an answer right away.’ […] *So then EventPlan was placed under Architecture’s control.*” [Developer, EventPlan]

The second mechanism, implementing technical workarounds, is a technical mechanism through which the team uses technology to postpone the need to resolve a dependency by working around it.

> “We had expected it to some extent. We had to implement a method. […] If someone needs to work on a … on the ETL object in a directory, he prefixed with his, we set up abbreviations for users… He prefixed the directory. *To make it as if we took the lock, finally.*” [QA Lead, LegalBI]

Technical workarounds may be simple or complex. They allow the team to work without suffering from the lack of participation of dependencies in the project. Notwithstanding, a drawback of technical workarounds is that they may involve rework at a later date, if the dependency is resolved and the workaround proves unfaithful to the process that must actually be followed.

> “So we carried on like that, except that each time it was delayed some more, and we sent pokes to try and get more information […]. Then we tried to work on other tasks or pretend that it was ready. […] *But of course it led to…, late decisions that meant reworking things a lot here and there.*” [Developer, ResourceMgmt]

The third mechanism, automating deployments, is a technical process enabling the autonomous delivery of software artifacts through automation to avoid dependencies.

> “It was really tough slogging to make builds and all the rest. I think that now it’s done more automatically, *we spend a lot less time on it. […] There are no real downsides.*” [IT manager, ScheduleMgmt]

At Logistics, software artifacts for testing and production environments were traditionally built and deployed by members of the technical team. However, this process was largely manual and error-prone. Indeed, in agile ISD, activities such as software deployments are considered complex and are best executed by automatic processes that can perform “daily deployments” (Beck & Andres, 2004) at night (Beavers, 2007) so that the artifacts are ready to be used in the morning. ScheduleMgmt was built and deployed automatically using scripted tasks that allowed developers to be largely independent from the technical team. This ensured that the artifact was delivered faster, more frequently, but also that developers and members of the technical team could focus on other, value-adding activities.

> “As for the tests, now we have lots of really intelligent, practical tools *that allow us to, with just a click, take the last production backup and deploy it in a test and all that. So it was…, we call it ‘the button.’ And [the developer’s] button is very, very highly appreciated.*” [Business analyst, ScheduleMgmt]

Resource elasticity (fixed human resources versus ***elastic human resources*****).** To address resource elasticity tensions we observed one mechanism. Delegating independent tasks to actors located outside of the team’s core is a social mechanism that enables the team to add or remove resources based on the need to perform tasks that are independent from the main development of the artifact.

> “We got organized to give him some parts that were a bit more substantial, with fewer dependencies. Q – *You mean things that he could develop on the side. R – Yes.*” [Project manager, LegalBI]

To address resource elasticity tensions, teams maintain a fixed core while allowing additional resources to participate in the project on a per need basis. However, this is only a viable option when those resources can work on tasks that have little dependencies with the rest of the project. For example, ResourceMgmt’s project manager ensured that new resources brought late in the project were exclusively focused on developing reports following guidelines from the main developers and were not involved in the development of the rest of the application. This mechanism alleviates some of the issues associated with the transfer of contextual knowledge in ISD projects (Dibbern et al., 2008; Levina & Vaast, 2008) and the onboarding of new resources. In line with this practice, agile ISD favors fixed-size teams that can reach outside of the team to address specific roadblocks. For example, Beck and Andres (2004) recall the case of a team hiring outside consultants for a specific need to allow the team to stay focused on its main objectives. From this perspective, resources remain fixed but independent tasks may be delegated to outside players.

3.4.2.3. Agile::Org Tensions – Learning

Mechanisms successfully addressing tensions that relate to learning are social mechanisms that are based on a conceptualization of ISD as a trial and error process where failure “imparts knowledge” (Beck & Andres, 2004:32). Our data suggest that learning occurs when teams (1) dedicate time to reflect on their efficiency and effectiveness while (2) limiting the exposure of the impacts of their failures with the rest of the organization.

Failure (opportunity for change versus ***undesirable event*****)**. We observed two main mechanisms addressing learning tensions. The first one is a combination of two mechanisms. Monitoring failure internally is a social mechanism that enforces regular periods of reflection on the efficiency and effectiveness of the team's work processes.

> "As for what we learn in the sprints, well, agility, it's part of the sprint review. […] In the specific case of MarketBI, one or two months went by before it came, but I gave it time, I trusted [the process]. They fell flat on their faces a couple of times, real bad, because they didn't have a peer review. *They were the ones who suggested that it be implemented. […] It wouldn't have been done that rigorously if the idea hadn't come from them in the first place.*" [Project manager, MarketBI]

The Agile Manifesto and agile ISD methods advocate self-reflection at regular intervals to help tune the ISD process throughout the project. Conversely, retrospectives or "post-mortems" are only conducted at the end of the project in plan-driven approaches. To generate the type of feedback necessary to trigger change, retrospective meetings (e.g., sprint reviews in Scrum) are performed at regular intervals and aim at providing a safe forum for team members to voice concerns and work on improving the ISD process. Our data suggest that this is achieved by ensuring that the lessons learned are kept within the team rather than communicated with the rest of the organization. For example, in ResourceMgmt the team's war room – the room where the team is collocated – was always opened to the public and a sign invited other employees to come to their "open house" every week. However, the war room's door always remained closed for the duration of retrospective meetings.

Limiting transparency is a mechanism we observed in combination with the internal monitoring of failure. As we have explained previously, limiting transparency allows the team to decide which information should be shared with the organization. With regards to failure tensions, limiting transparency allows the team to implement changes without having to communicate their failures with the rest of the organization. Together these two mechanisms provide a space for reflection shielded from outside criticisms.

> "But *it wouldn't have been done that rigorously if the idea hadn't come from them in the first place*. Then again, was I telling the client that they had lost 5 days because we fell flat on our faces? Absolutely not, I didn't say a word." [Project manager, MarketBI]

The second mechanism, enacting symbolic performances, has also been described previously in the context of driving force tensions.

> “Our sprint reviews were so poor. They were completely worthless […]. The scrum master would lead it in a way that it would be positive, of course. So, you doodle a bit and draw happy faces… [...] *I don’t know whether they suffered really [...] but I tend to think that [the developer] suffered in silence.*” [Business analyst, EventPlan]

In three out of five cases (EventPlan, ResourceMgmt, and LegalBI), the enactment of sprint reviews and other retrospective meetings was largely symbolic. While the need for change was acknowledged and included in written documentation (e.g., in ResourceMgmt, the lessons learned from every sprint review meeting were compiled in a written document), responses to this need for change were rarely enacted. Retrospective meetings would consist in building bullet lists of “lessons learnt” for future projects rather than the current one. In LegalBI, this led to some unresolved quality issues that were brought up by the client to the team. While those changes resulted in positive outcomes (increased artifact quality), they were not the result of a reflection effort. Rather, they were caused by pressure from outside of the team.

> “A sprint review, well it has two parts: first there’s the team, then there’s the project itself. […] *We made some adjustments. I’m not saying that there weren’t any adjustments, but it wasn’t systematized. Yes, some adjustments were made, but not with as much care as they could have been* compared to, or is it…, but that’s typical, I see it often enough. The sprint planning goes well, or relatively well, but the sprint review *is often a bit flaky, it’s so-so.*” [Project manager, LegalBI]

> “Now really, the client was … 'You’re producing garbage… That’s unacceptable.’ […] So we said, alright, here’s a good opportunity for a sprint review, so we reorganized […] *to improve things.*” [Delivery manager, LegalBI]

In ScheduleMgmt, no time was allocated to perform retrospective meetings because the team was constantly trying to catch up with the project’s deadlines. As a result, the need for change, while acknowledged by the VP of product development, was dismissed by the President as a short-term issue, hindering developer satisfaction and maintaining the team in a precarious situation.

> “And they were brilliant, but you know, *the [architect] who was so upset, it’s tough to see that happen*. To think that I pushed the guy that far. And it comes down to us, it’s our fault: me, [the project portfolio manager], [the project manager] and [the president]. You know, it’s people who manage that, *and we didn’t do our job that time. We really got ourselves in trouble at that point.*” [VP product development, ScheduleMgmt]

3.4.2.4. Agile::Agile Tensions – Leanness

Mechanisms successfully addressing tensions that relate to leanness were, in all but one instance, technical mechanisms. These technical mechanisms aimed at (1) supporting value-adding activities (e.g., building a solid architectural foundation) and (2) automating non value-adding activities in such a way that delivering the software artifact is simple, economical, and enforces high levels of quality. Consistent with the principles of agile ISD, this must be done quickly and at regular intervals so that users can provide timely feedback to the team. In the sections below, we present those mechanisms as well as their impacts on projects.

Pace and extrinsic quality (speed versus *extrinsic quality***)**. We have observed two mechanisms that address tensions related to pace and extrinsic quality. The first mechanism, exclusively enrolling business, is the team's exclusive reliance on users and the rest of the organization to assess system quality.

> "You have to get things aligned with the client and say, 'Well, we'll be delivering something each week, but you need to have people on your end or it won't be worth the bother.' We did that work, and the client said, 'Yes, yes, yes,' *but in the end, they always spread themselves too thin. […] If he says yes, but then doesn't follow through and tries […] They got on board a little late.*" [Business analyst, MarketBI]

In the prescriptions of agile ISD methods, users are committed to the project and are able to provide timely feedback to developers. System quality is assessed continuously, changes are implemented quickly and validated during the course of the next iteration. Our data suggest that users and external proxies were often unable to provide the timely feedback required to enable this process due to commitment tensions. Rather, they performed quality assurance at irregular intervals, which prevented the teams from integrating this feedback in a progressive manner. Underlying commitment tensions therefore resulted in coordination issues for the team, hindering the pace of development.

> "Yes, we do deliver quickly, in two weeks we had things ready to be tested on the client's system. […] We were waiting on the test results, but you know, we had already started other iterations. So there was that package to be tested and *the power users didn't necessarily have the time to do it, so it may not have really been a priority to come support us at the speed we were moving ahead. Such that the tests were delayed, and delayed, which added to our work load, after the fact, once they decided to do their testing.*" [Developer, MarketBI]

The second mechanism, automating tests, is the use of automated tests to validate large parts of the artifact's extrinsic quality without requiring any intervention from users or proxies.

> "Well, it's my initiative to allow continuous integration, to have all our infrastructure. We commit in Subversion and it's compiled, *and we know right away whether or not it compiles* and it executes tests." [Developer, ScheduleMgmt]

In agile ISD, the automation of tests is often linked to the practice of test-driven development – TDD – (Sanchez et al., 2007) advocated by XP. This practice consists in writing a failed test case first and then working to write the code that will eventually make the test case pass. TDD reverses the traditional quality assurance process wherein test cases are written after the code is developed. One of the main purported benefits of TDD is that it generates a battery of test cases that can be automated during the build process. While none of our cases performed TDD, two of them relied on small unit tests akin to those used in TDD to automate part of the quality assurance process.

> "There were tests that were coded on the server side. […] Since oftentimes the major business rules are on the server side. It's essential, you know." [Developer, EventPlan]

Pace and intrinsic quality (speed versus *intrinsic quality***)**. We observed three mechanisms contributing to address pace and intrinsic quality tensions. The first mechanism, Limiting transparency, allows the team remain shielded from the rest of the organization in order to bring changes to the artifact's intrinsic quality as part of its regular workload.

> "The number of workshops that we had on the side, on the corner of a table, 20 minutes to discuss the concept we were going to implement and to challenge each other on it. […] You hide that from the client big time. Big time. […] *[Agility] allows you to make decisions without consulting the client, without him really noticing it. It's for his own benefit*." [Project manager, MarketBI]

Consistent with our previous observation of this mechanism to address driving force tensions, our data suggest that in MarketBI, limiting transparency allowed the team to make choices on the artifact's internal structure without having to worry about the opinions of members located outside the team.

The second mechanism, anticipating decisions, is a social mechanism where developers anticipate decisions in order to maintain their velocity even if they are waiting for dependencies to be resolved.

> "That's when we tried to either work on other tasks or *pretend that it was ready*… Simulating what we expected in terms of results, in any event." [Developer, ResourceMgmt]

In ResourceMgmt, the team used simulation to anticipate some decisions from the architecture team and maintain its pace. Unfortunately, while this helped the team advance in an autonomous manner, the resolution of dependencies induced significant rework as the decisions anticipated by the team did not match those eventually taken by dependencies.

> “When we finally received word from Architecture, ‘OK, here’s how it’s going to work,’ I had done a sprint, a sprint and a half to tear down our database model again and then rebuild it. […] *This is work with a rather limited ROI.*” [Developer, ResourceMgmt]

> “Well, we had to create an environment that somewhat simulated a real environment, but was not exactly the real environment. And deal with the resulting imperfections. […] *It would end up causing some problems. We would discover things that, in the end, we weren’t able to test. The testing capacity was reduced. Especially with the integrated tests.*” [Project manager, LegalBI]

> “Actually, we hadn’t planned on using OData. At first, we thought that the message bus would be enough, but finally, no, *we realized that the [other project], in fact, that they need to consume, like batches of data, and interfaces needed to be developed.*” [Developer, EventPlan]

The third mechanism, exploring architectural patterns, is a technical mechanism through which the team dedicates time, either at the beginning of the project or at regular intervals, to evaluate the architectural choices that must be made throughout the development of the artifact.

> “And in the beginning, it was Sprint 0, we created compliant dimensions. I looked at all these models and thought, ‘Oh, my God. Will there be one dashboard? Five? A dozen? How will they communicate? How will it…?’… *So, understanding the basic outline of the solution’s architecture*. […] A blueprint solution, we call it.” [Director of practice, MarketBI]

Establishing a solid software architecture is time-consuming (Boehm & Turner, 2005). In addition, refactoring implies that developers are able to reflect on potential improvements to the reusability and readability of the code they write (Opdyke, 1992). In line with past research, our data suggest that combining speed with intrinsic quality was most easily achieved when team members were highly skilled (Erickson et al., 2005). The reality of four out of five of our cases is that the skill level of team members varied greatly. As a result, having a technically-oriented iteration (Sprint 0) where no working software was delivered (Hughes, 2008) helped evaluate the required level of intrinsic quality of the artifact under development.

Automating deployments, which we have described previously, is the third mechanism that addresses pace and intrinsic quality tensions.

> “So Maven arrived at the same time as Hudson to make our automatically compiled packages. At the same time, it does our automatic tests and automatic builds. We wrote a lot of scripts to automate it all. So now we have scripts in Hudson to deploy to new clients, to update databases, update clients. *The end result today is that it was also done so that programmers no longer need to update clients, it could be someone in implementation who can say, ‘OK, I’ll test such-and-such a client, I’ll update such-and-such a version,’ and it’ll be done automatically.”* [Developer, ScheduleMgmt]

With the advent of virtualization technologies and cloud computing platforms, organizations can easily allocate computing resources. In ISD projects, this means that several aspects of the hardware and software environment where artifacts are deployed and that used to be outside of the team's control can now be shaped by the artifact's requirements and be modified at a later date if necessary. For instance, cloud computing allows organizations to replicate with a great degree of accuracy complex production environments inside test environments. At Logistics, this meant that all of the software components developed were standardized so as to be built and deployed using a standard procedure. Creating and configuring client environments was automated and never required the intervention of the technical team.

> "*If you wanted to add a client, it was really a lot of work*. So, after that we modified it so that it would be closer to what we wanted. Because the Logistics method is a bit like that, it's a bit like with the *need to have deployment capability*…" [Architect, ScheduleMgmt]

Similarly, Entertain also relied on cloud-based environments for the development of EventPlan and ResourceMgmt. However, deployment automation was never attempted and significant differences existed between test and production environments. These factors rendered releases in production manual and error-prone. In addition, Entertain had added rather than removed a layer of dependency with the use of a third party provider in charge of production deployments. This led to an environment where deployments were only partially automated in test environments, creating unexpected delays and coordination issues between the team and third party providers during production releases.

> "[Third party] is like a puppet under remote control. They have no expertise and do the job. […] *We lost a lot, a lot of time on that, and we're still losing time today*. But apart from that, yes, we could have had all the tools to easily launch the builds and all the rest, it's just that the method used at the end was really…, that's it, we had to go through [third party]." [Developer, ResourceMgmt]

Pace and communication (speed versus *communication***)**. We have observed two mechanisms to address pace and communication tensions. The first mechanism combines Performing short, frequent status update meetings and integrating collaboration technologies. Performing short, frequent status update meetings, is the use of short meetings taking place at regular intervals.

> "I was telling people, 'OK, offline, it's finished. Let's move on.' *to try and keep things moving because otherwise it was the kind of scrum that could easily stretch into an hour.*" [Director of practice, MarketBI]

In Scrum these short meetings are called daily scrums (Schwaber & Beedle, 2003:3) and require the communication of three elements from each participant: (1) what they did since the last Daily Scrum; (2) what they plan on doing after the Daily Scrum; and (3) whether they are experiencing any roadblocks. In the Scrum tradition, those meetings are conducted standing up to ensure that they remain short and are moderated by a Scrum Master. In all cases except ScheduleMgmt, daily meetings were performed. In EventPlan, the meeting was moderated using a small stress ball passed around to indicate who had the right to talk. In agile ISD projects, communication is strongly encouraged but it also requires discipline to ensure that it does not impede on the team’s velocity.

> “The daily scrums, you’ve got to maintain them. […] Agility, *it helps break down the cubicle mentality, it’s a lack of cubicles, so people talk with each other.*” [Project manager, MarketBI]

> “There’s a communication mechanism that gets established which is not a free-for-all in which, if I need someone, I’ll just go to his office. No, no. ‘Do you have some time right now?’ *Because otherwise I may ruin another deliverable. You’re concentrating on something you’re doing and I break in with my thing*. So sometimes people forget that, and think that agility means that everyone is talking all the time.” [Project manager, MarketBI].

Integrating collaboration technologies, is a technical mechanism that enables synchronous and asynchronous communication between team members regardless of their physical location.

> “There’s always a bridge. Yeah, it’s part of what I’d call the basic tools, it’s the bridge, *accessibility and all that. So no-one had any more excuses*. […] *In fact, the technical part of access to the scrum was never an issue*.” [QA Lead, LegalBI]

These technologies allow team members to actively participate in meetings even if they are away. In all our cases, projects were managed using tools (e.g., JIRA) that integrated work items, documentation, design documents and discussion threads on those work items among team members. During meetings, remote participants used synchronous communication technologies (e.g., conference call, video conference, screen sharing) and had access to those tools to quickly access documents.

In agile ISD research, Sarker and Sarker (2009) have noted the importance of technology to enable agility in distributed ISD teams as well as the role of short, frequent meetings to ensure a smooth transition of work across time zones. Our analysis reveals that the combination of these two mechanisms allowed the team to maximize the efficiency of communication within the team while ensuring that the information was properly disseminated.

> "For us, *for project management* we use JIRA for everything related to new functionalities, bugs, improvements, tasks, that sort of thing, to get the job done, you know. When more documentation is needed in JIRA, there will be a link to a wiki. We have a wiki that documents more complex things, and *there is more information to which JIRA can be attached, from which it can draw.*" [Developer, ScheduleMgmt]

Notwithstanding the perceived usefulness of collaboration technologies, our data suggest that external proxies and members from the organization rarely used them. While all of the project's information was stored within those tools and integrated across them, only team members were using them on a regular basis.

Using physical artifacts is the second mechanisms we have observed to address speed and communication tensions, especially with external stakeholders (e.g., external proxies).

> "The things for the wall, *they're for people from outside*. We don't care, we don't even look at them. That's really all they're for. *When people walk in, they're able to read what's on the walls and they see the progress we've made. It's more for the bosses*. TFS, that's really where we were monitoring our tasks." [Developer, ResourceMgmt]

Physical artifacts (e.g., whiteboards) are one of the most important elements advocated by agile ISD methods (e.g., Beck & Andres, 2004) to enable face-to-face communication and quickly provide a high level overview of the project's progress. In LegalBI and ResourceMgmt, physical artifacts allowed external stakehoders (e.g., users, external proxies) to quickly gain an understanding of the project's progress without disrupting the rest of the team. However, our data also suggest that in some cases (LegalBI), using physical artifacts gave rise to role tensions if external stakeholders manipulated the artifact while the team perceived it as their responsibility.

> "We didn't even have [whiteboard] at the start, we just installed it at one point and it's great to have, it's a great tool. *Even the client can use it to see our progress. And there were people on the project who weren't there every day. So you work hard for two days, you're away for three days, and when you get back you can see things right away, where we are, what's been added, what's been accomplished*. […] *The product owner was in front of the board and moving post-its around for us, saying, 'OK, this is a priority. I'm putting it here. That's done, so I'm showing it as done'. […] You know, at times it was a bit intrusive and difficult to keep in check*." [Developer, LegalBI]

3.5. Experiencing Agility in ISD Projects through Tensions

Our three research questions focus on (1) the challenges experienced by teams in agile ISD projects, (2) the mechanisms that serve to address those challenges, and (3) the impacts of those mechanisms on projects. In this section, we build on the findings that emerged from the iterative analysis of our data to offer a theoretical explanation answering those questions, presented in Figure 3.1.

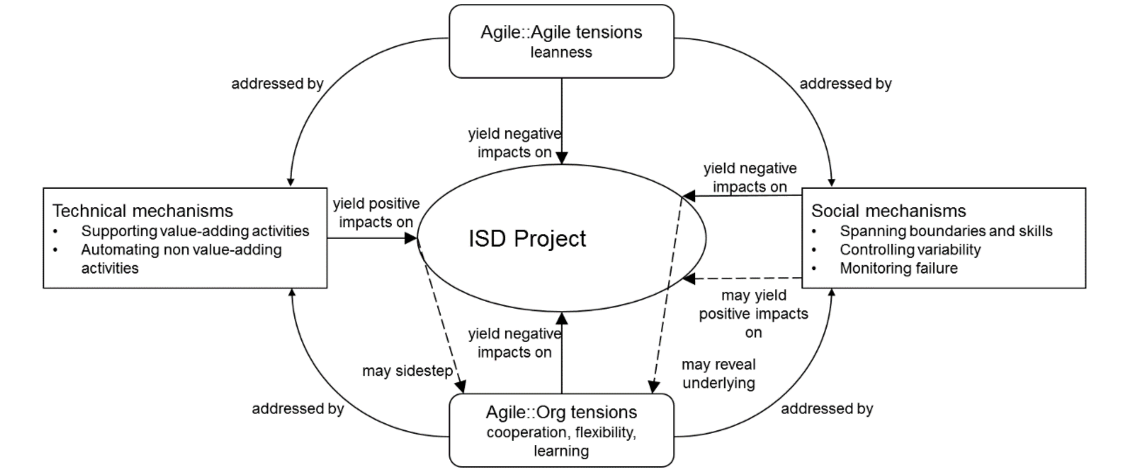

Agile::Agile tensions
leanness
addressed by
addressed by
yield negative impacts on
yield negative impacts on
Technical mechanisms
• Supporting value-adding activities
• Automating non value-adding activities
yield positive impacts on
ISD Project
Social mechanisms
• Spanning boundaries and skills
• Controlling variability
• Monitoring failure
may yield positive impacts on
yield negative impacts on
may sidestep
may reveal underlying
addressed by
addressed by
Agile::Org tensions
cooperation, flexibility, learning

Figure 3.1. Experiencing agility in ISD projects through tensions

Our explanation identifies tensions, defined as “elements that seem logical individually but inconsistent and even absurd when juxtaposed” (Smith & Lewis, 2011:382), as central elements of agile ISD projects. Consistent with this view, we posit that elements within agile ISD projects such as the principles of the Agile Manifesto (Beck et al., 2001), practices prescribed by agile ISD methods or decisions originating from the organization mandating the development of a software artifact may be sensible when taken individually. However, the combination of two such elements may present an inconsistency that induces a tension. For example, in line with the principles of the Agile Manifesto, teams working in agile ISD projects should strive to build the “best architecture” and deliver working software at regular, short intervals. However, it has been argued that these two objectives may run against each other because software architecture takes time and planning (Boehm & Turner, 2005), representing a tension observable within an agile ISD project.

While research has studied challenges and lessons learnt in agile ISD projects at length (Dingsøyr et al., 2012; Dybå, 2002), this has almost always been done from the perspective of the adoption of agile ISD methods (Abrahamsson et al., 2009; Conboy, 2009). For instance, several works have argued that the adoption of those methods is hindered by an organizational culture that is not receptive of the prescriptions that they provide (e.g., Iivari & Iivari, 2011; Kim & Ryoo, 2012). Indeed, the authors of Scrum argue that the adoption of their method triggers a process of change in an organization's culture which is complex and spreads over long periods of time (Schwaber & Beedle, 2003). Our explanation approaches the agile ISD phenomenon from a different angle by focusing on the agile ISD project as its unit of analysis, regardless of any intention to adopt an agile ISD method or spread the use of an agile ISD method outside of the boundaries of a given project. As a result, our explanation does not posit that an organizational culture can run *against* the principles of agility in ISD. Rather, it studies the inconsistencies between elements of agility in ISD and the environment where the project is conducted. Explaining the experience of agility in ISD projects by studying tensions represents a pragmatic take on agility in ISD and contributes to answer calls to avoid studying the ISD phenomenon through the prism of methods (Truex et al., 2000).

Our model makes a distinction between two main categories of tensions. The first category, Agile::Org tensions, emerges from inconsistencies between principles of agility in ISD and the organization where the project takes place. For example, one of the key elements of agility in ISD advocated by agile ISD methods (e.g., Scrum, XP) is the complete commitment of the team to the project. In agile ISD, users represent important actors that drive the development of the artifact and whose daily participation is crucial to project success. Yet organizations may not be able to let users stop their daily tasks and work exclusively on the project. As a result, securing the exclusive commitment of users to an agile ISD project is challenging in many settings (e.g., Conboy & Fitzgerald, 2010; Strode, 2015).

Agile::Org tensions pertain to three of the four facets of agility: cooperation, flexibility and learning. The example of the tension between the full commitment of users and their partial commitment to the project pertains to cooperation because it relates to communication and the coordination of work during the course of the project. Tensions pertaining to flexibility emphasize inconsistencies between elements that affect the team's ability to adapt based on unforeseen contingencies. For instance, autonomy is an important principle of agility in ISD because it allows the team to be independent in the way it organizes itself and in the way the artifact is developed. Yet in a typical organization, software artifacts are often developed to integrate with one another, creating dependencies (e.g., developers must work according to enterprise architecture guidelines) that are at odds with the idea of autonomy found in agile ISD. Finally, tensions pertaining to learning are based on inconsistencies between the perception of ISD as a trial and error process where team members learn from failure and the perception that failure is an undesirable event that costs resources to the organization.

The notion of Agile::Org tensions suggests that several tensions observed in agile ISD projects emerge from inconsistencies with elements of the organization where the project takes place. More specifically, our model suggests that within the team, cooperation, flexibility and learning occur without friction. It is when teams interact outside of their direct boundaries that tensions pertaining to those three facets of agility in ISD emerge. Those tensions reveal some of the implicit assumptions that agile ISD methods have long ignored but that research on agile ISD has hinted at by studying factors such as organizational culture as hinderers of the adoption of those methods (e.g., Iivari & Iivari, 2011). For instance, many of the examples recounted by the authors of Scrum and XP as illustrations of their prescriptions are based on small, collocated projects staffed with a task force made of experienced developers and a customer fully committed to the project. Budgets and other constraints placed by the organization on the project are seldom discussed. Agile::Org tensions refute some of those assumptions by placing the project within a given organizational context, rendering those assumptions explicit.

The second category of tensions, Agile::Agile tensions, emerges from inconsistencies among principles of agility in ISD. Those tensions pertain to leanness, the fourth facet of agility in ISD. One of the key principles of agility in ISD that is rooted in its iterative and incremental approach to ISD is the speed at which the artifact is developed and delivered to users. This pace is considered instrumental to enable the rapid feedback loops that allow the software to grow quickly while becoming more and more in line with the needs of users. Yet this pace may be difficult to maintain when one seeks to develop software using principles of economy, simplicity and quality. For example, decisions made at the beginning of an iteration may need to be revisited at the beginning of the next iteration, causing rework and waste that go against economy. Similarly, working fast may run against the ability for the team to (1) develop an artifact using a solid architectural foundation or (2) ensure an appropriate level of system quality, one of the most important dependent variables in IS (DeLone & McLean, 1992).

Agile::Agile tensions provide an alternate perspective on the agile ISD phenomenon and suggest that tensions are still present in environments where an organization fully embraces the principles of agility in ISD. Research on agile ISD has traditionally focused on the adoption (e.g, Weiyin et al., 2011) and customization (e.g., Conboy & Fitzgerald, 2004) of agile ISD methods and identified the context within which those methods are applied as an important source of challenge. Within this stream of literature, the Agile::Org category of tensions described previously is relevant. Agile::Agile tensions shift the focus to the principles of agility and the prescriptions of agile ISD methods by acknowledging that they may be inconsistent with one another. For example, the fast pace imposed by the incremental and iterative features of agile ISD methods and agility in ISD in general are difficult to reconcile with the need to create a solid architectural foundation that can support the evolution of the artifact over time. Agile::Agile tensions therefore draw attention to the fact that there are contradictions within the principles of agility in ISD that can be problematic even in cases where Agile::Org tensions are absent.

In the presence of a tension, three scenarios can occur. The first scenario represents a situation where no mechanism serving to address the tension is observed, either because the tension was never acknowledged or because nothing was done to address it. In this scenario, tensions are exacerbated and the project is negatively impacted. The second scenario involves social mechanisms, which we define as series of activities that are founded upon interactions among team members and between the team and the rest of the organization. The third scenario involves technical mechanisms, which we define as series of activities that are founded upon the use of technology and tools to address tensions. While past research has offered standard *responses* to tensions (Van de Ven & Poole, 1988), these responses have been conceptualized as cognitive processes located at the individual level and are therefore assumed to be intentional. In contrast, our work proposes mechanisms at the project-level without inferring on the intention that may drive their enactment. For instance, certain activities may be put in place as a result of agile ISD method prescriptions (e.g., Scrum rituals) while others may aim at addressing a specific tension once it is acknowledged by the team. Overall we suggest that these mechanisms may involve activities performed by one or many actors involved in the project.

As illustrated in Figure 3.1, social mechanisms can serve one of three purposes. First, they can help team members span across the boundaries of the project and the skills of team members. For example, the team may designate a *proxy* who acts as an interface between users and developers in situations where users cannot be committed to the project on a full time basis. Second, they can serve to control variability to ensure that variations occur without compromising the project's goals. For example, a project manager may allow the scope of the project to fluctuate but only within the constraints of the budget set by the organization at the onset of the project. Third, they can serve to monitor failure so that mistakes or issues are perceived as opportunities for learning and change rather than undesirable events. For example, team members may reflect upon their mistakes at the end of an iteration to change their work processes while hiding those mistakes from the rest of the organization so as not to report on them as costly failures.

Technical mechanisms can serve one of two purposes. First, they can support value adding activities of the ISD process. For example, the prescriptions of agile ISD methods and the literature on agile ISD have emphasized the role of communication (e.g., Pikkarainen et al., 2008) in agile ISD projects. To that end, short, frequent meetings such as daily standup meetings (Schwaber & Beedle, 2003) are recommended. However, in situations where commitment tensions are present, those meetings may not always be achievable. In those instances, the presence of tightly integrated collaboration technologies help team members communicate in an asynchronous manner and support synchronous communication processes. Second, technical mechanisms can automate non-value adding activities. For example, tensions that pertain to leanness emphasize the importance of the pace of delivery. Automated build and deployment systems can automate a lot of the tasks involved in the delivery of a software artifact and give team members more time to focus on other, value-adding activities.

While social and technical mechanisms can serve to address both Agile::Org and Agile::Agile tensions, we suggest that their impacts on the project may vary. Because social mechanisms target interactions among actors, they relate to the first three facets of agility in ISD that emphasize those interactions: cooperation, flexibility and learning. As a result, we posit that Agile::Org tensions are best addressed by social mechanisms. While technical mechanisms can also contribute to address Agile::Org tensions, they do so by supporting value-adding activities that rely on interactions among team members and between the team and the rest of the organization. For example, short, frequent status update meetings – a social mechanism – cannot fully address commitment tensions because actors may not always be physically available when needed. The addition of integrated collaboration technologies – a technical mechanism – helps to support this social mechanism and allow actors to participate in those meetings remotely and share information with team members asynchronously (e.g., using a project management tool that enables discussion threads on work items).

In contrast, Agile::Agile tensions are best addressed by technical mechanisms that support the achievement of leanness. Indeed, we argue that those mechanisms can help automate many non-value adding activities and contribute to increase the pace of delivery while allowing the team to focus on the development of the artifact. For example, automated tests and automated deployments perform quick validations of the intrinsic (e.g., architecture) and extrinsic (e.g., perceived system quality) quality of the artifact and deliver it to users in an unattended manner. The emergence of virtual and cloud-based computing platforms allow ISD teams to quickly deploy resources that are replicated between test and production environments. As a result, enacting practices advocated by agile ISD methods such as XP's continuous integration is not only feasible, but is achievable with little overhead for the team. Conversely, we argue that social mechanisms addressing Agile::Agile tensions can have a negative impact on the project because they increase the number of interactions among actors. For example, delegating deployment tasks to a third party may initially save time for team members. However, the complexity of deployments may cause an increase in communication and coordination between the team and the third party in the long run, causing delays.

Consistent with the literature on tensions, we suggest that mechanisms yield successful outcomes on projects when they create a virtuous cycle where both elements of a tension are reconciled and a compromise is achieved (Smith & Lewis, 2011). For example, allowing the scope of a project to vary while always remaining within budgetary constraints ensures that users can select and prioritize features within the limits set by the organization. If only one of those elements were favored over the other, one may incur budget overruns or user frustration. In situations where one element of a tension is favored over the other, negative outcomes are obtained. For example, following Scrum practices in a symbolic manner to give the impression that users drive the development of the artifact while letting the organization effectively control it favors the organization as the main driver of development and can generate frustration within the team and with users. It is therefore not only sufficient to rely on social or technical mechanisms based on the category of a tension and the facet of agility in ISD to which that tension pertains. Our explanation goes beyond the logic of opposition traditionally put forward in studies on agile ISD adoption or the prescriptions of agile ISD methods: to enable virtuous cycles, mechanisms must trigger a process wherein the complementarity of the elements of a tension are sought rather than the substitution of one over the other.

Finally, our model conceptualizes tensions as interdependent rather than singular objects in the context of agile ISD projects. While tensions represent inconsistencies between two elements, we argue that mechanisms that serve to address those tensions may reveal other, underlying tensions. For example, proxies are an important mechanism to address commitment tensions that pertain to cooperation in agile ISD projects. To act as interfaces between the team and users, proxies must possess both technical and business skills. When proxies selected outside of the team (i.e., *external* proxies) step outside of their expected responsibilities because they do not possess the necessary technical skills to interact with the team, other tensions related to their role within the project emerge.

Our explanation suggests that the changes in patterns of interactions induced by social mechanisms may render salient latent Agile::Org tensions. Indeed, while social mechanisms may contribute to address salient Agile::Org tensions, they may not be well-suited to address Agile::Agile tensions. Conversely, technical mechanisms can automate many of the testing and deployment tasks involved in the delivery of a software artifact. They may therefore remove some of the complexity and frequency of interactions among actors involved in the project and sidestep the emergence of latent Agile::Org tensions. Our explanation accounts for the complex and interdependent nature of tensions and emphasizes the idea that underlying tensions may become salient when mechanisms trigger changes in the patterns of interactions among actors.

3.6. Conclusion

In this work, we asked the question “how do teams experience agility in ISD projects?”. Specifically, our work aimed at studying: (1) the challenges experienced by teams in agile ISD projects; (2) the mechanisms that serve to address those challenges; and (3) the impacts of those mechanisms on projects.

Our review of extant literature on the topic led to two main observations that motivated our work. First, research on agile ISD reflects an implicit assumption equating the adoption of agile ISD methods with the experience of agility in ISD. We observed that this assumption fails to account for the mechanisms that can help to address some of the factors inhibiting agility in ISD as a means to conduct an ISD project. Second, the literature emphasizes the role of interactions among actors as the main manifestation of agile ISD (e.g., Maruping et al., 2009; Strode et al., 2012). While in line with many practices prescribed by agile ISD methods, we suggested that this propensity fails to account for the role technology can play to help teams address challenges and support agility in ISD projects.

Aiming at filling those gaps and building a theoretical explanation, we adopted a multiple case study design. We collected data in four ongoing and one retrospective projects where agility was perceived as an important factor for success in three different organizations. Using coding techniques borrowed from grounded theory, we identified *tensions*, defined as inconsistencies between “elements that seem logical individually but inconsistent and even absurd when juxtaposed” (Smith & Lewis, 2011:382) as important elements creating challenges in those projects. We then proceeded to study the mechanisms that served to address those tensions and the impacts of those mechanisms on projects. Building on those elements, we developed an explanation on the experience of agility in ISD projects through tensions.

Our work makes four contributions to the literature on agile ISD. First, we propose two categories of tensions in agile ISD projects: (1) Agile::Org tensions between principles of agility in ISD and the organization and (2) Agile::Agile tensions among principles of agility in ISD. While Agile::Org tensions encompass challenges experienced in agile ISD projects and reported in past literature (Boehm & Turner, 2005), they do so without considering agile ISD method adoption as an overarching objective of teams. In addition our conceptualization of tensions does not posit that one element of a tension runs against the other. Rather, we suggest that tensions are inherent to ISD projects and that the inconsistency between their constituting elements represent challenges for teams working to develop a software artifact. The identification of Agile::Agile tensions also suggests that tensions are relevant even in settings where the organization was designed to support agile ISD (e.g., Kniberg & Ivarsson, 2012). Future research may further study Agile::Agile tensions to understand their implications on ISD projects, e.g., in scenarios where teams may have to choose one agile element over the other due to certain constraints imposed upon them.

Second, we extend the literature on agile ISD and account for the instrumental role that technology can play to address tensions in ISD projects. While past research has studied many of the practices prescribed by agile ISD methods revolving around interactions among actors, the role of technology in helping teams deliver artifacts quickly is still largely unexplored. Our explanation suggests that social mechanisms aiming at spanning boundaries, controlling variability and monitoring failure are best suited to address Agile::Org tensions. In addition, technical mechanisms aiming at enhancing a team's velocity by automating tasks and supporting communication among actors help to address Agile::Agile tensions because they ensure that the team can work using principles of economy, simplicity and quality. Future research may study in more detail the evolving role of technology as an enabler of agility in ISD as new technologies allow for faster development and deployments of software artifacts. For example, Amazon recently launched AWS CodePipeline (Amazon.com, 2015), a cloud-based continuous delivery system that builds, tests and deploys code automatically. Another avenue for future research is the distinction between intentional and accidental mechanisms put in place to address tensions. Indeed, in some instances, our data show that mechanisms were purposefully put in place to address tensions by teams while in other cases, practices were enacted for a number of other reasons (e.g., to follow prescriptions of an agile ISD method) and still helped to address tensions when they became salient. Future research on the topic may help us gain a better understanding of the ability for mechanisms such as those derived from prescriptions of agile ISD methods to preemptively address tensions.

Third, we theorize on the impacts of mechanisms contributing to address tensions in ISD projects. Consistent with works on tensions and paradoxes (Lewis, 2000; Quinn & Cameron, 1988; Smith & Lewis, 2011), our explanation suggests that positive outcomes are observed when mechanisms aim at achieving a compromise where the two elements of a tension are taken together. While this entails that those elements may not each be maximized within the context of the project, it accounts for the reality of many ISD projects where constraints prevent them from doing so. Our work offers a different take on this reality by presenting compromise as a preferable scenario over the favoring of one element over the other to maintain a balance between the elements of a tension.

Future works may conceptualize agility as a means to conduct an ISD project rather than an overarching objective for an organization and uncover more nuanced insight with regards to the suitability and use of agile ISD methods in organizations (Boehm & Turner, 2004). Future research may also look beyond overall positive or negative impacts and study the nature of the impacts of mechanisms contributing to address tensions in ISD projects. For example, certain mechanisms (e.g., the integration of collaboration tools) may foster better coordination among actors while others may help the team deliver the system faster (e.g., automated deployments). Having a better understanding of this relationship may help teams prioritize certain mechanisms over others based on the objective they seek to achieve.

In addition, we uncover the interdependencies that may exist when social mechanisms render Agile::Org salient. Indeed, social mechanisms rely on interactions among actors. We suggest that these mechanisms can change patterns of interactions within the project and reveal underlying Agile::Org tensions that were latent until those changes rendered them salient. This argument further anchors the relevance of technical mechanisms to address Agile::Agile tensions. As a result, considering tensions as independent objects that can be addressed by a single mechanism hides an important part of their complexity. Future work may study tensions from a longitudinal perspective to better understand the *process* of emergence of tensions throughout the course of a project.

Fourth, our specification of agility in ISD makes a minor contribution to the literature on agile ISD. We have proposed four facets of agility in ISD that may serve as a basis for future research seeking to refine the conceptualization and operationalization of the construct of agility in ISD. While past research has offered research-based definitions of agility in ISD, it has been observed that most of those definitions still rely on the features of agile ISD methods (e.g. Abrahamsson et al., 2002). In other cases, those definitions are built on definitions found in other fields (e.g., Conboy, 2009) and still lack empirical validation. Our work builds on extant literature in agile ISD to propose cooperation, flexibility, learning and leanness as overarching facets of agility in ISD, reflecting the multifaceted nature of this concept (Sarker et al., 2009).

Finally, our work makes a methodological contribution to qualitative research. The design of structured data analysis techniques to formalize the organization of evidence coded using principles borrowed from grounded theory allowed us to reinforce the reliability of our analysis. By rendering the links between our raw data, preliminary chains of evidence, tensions, mechanisms and impacts explicit using technological artifacts specifically designed for this work, we were able to quickly assess emergent findings and contrast and compare those findings across our cases while ensuring traceability. Future works may use some of the techniques described here to help analyze rich, unstructured qualitative data.

This work also has some limitations that affect the generalizability of our findings. First, while some team members were working remotely because they were assigned to multiple projects, projects were, for the most part, collocated. Therefore, it would be interesting to replicate our findings in distributed or ISD offshore settings. While we have no data to draw any conclusions, one may expect that technology would play an even more important role as it mediates much of the communication among team members in these contexts. Second, the projects where agile ISD methods were used all relied on Scrum. While extremely popular, there are other methods such as XP and more recently, DevOps (Kim et al., 2014), that have placed more emphasis on the role of technology to enhance teams' velocities. Studying projects where those methods are used may reveal additional insight on the role that technology can play to support agility in ISD.

Notwithstanding those limitations, our work has implications for practitioners. First, we suggest that tensions are always present in ISD projects. Rather than looking to adopt agile ISD methods at the scale of the organization, practitioners may consider agility as a means to execute a given project and use those means to address tensions with the rest of the organization and within the principles of agility in ISD. Second, our work emphasizes the role that technology can play to address Agile::Agile tensions and enhance a team's velocity. While those mechanisms may be costly to implement initially, the rapid evolution of technology and the availability of cloud computing platforms supporting those technologies means that the ability to incorporate those mechanisms as part of a team's workflow is becoming more cost-effective. In addition the standardization of those technologies can help streamline processes that may differ across projects by enforcing common patterns for the development and deployment of software artifacts. Finally, our work points to the need to identify relationships between salient and latent tensions. When selecting mechanisms to address salient tensions, teams must take into account the possibility that those mechanisms can render other, latent tensions salient.

References

ABRAHAMSSON P, CONBOY K and XIAOFENG W (2009) 'Lots done, more to do': The current state of agile systems development research. *European Journal of Information Systems, 18*, 281-284.

ABRAHAMSSON P, SALO O, RONKAINEN J and WARSTA J (2002) Agile software development methods: Review and analysis: VTT Technical Research Centre of Finland.

AMAZON.COM (2015) AWS codepipeline | continuous delivery. https://aws.amazon.com/codepipeline/, accessed February 2, 2016.

ANTI AGILE (2015) The anti agile manifesto. http://antiagilemanifesto.com/, accessed January 6, 2016.

AOYAMA M (1998a) Agile software process and its experience. In *International Conference on Software Engineering* pp. 3-12, Kyoto, Japan.

AOYAMA M (1998b) Web-based agile software development. *IEEE Software, 15*(6), 56-65.

BALIJEPALLY V, MAHAPATRA R, NERUR S and PRICE KH (2009) Are two heads better than one or software development? The productivity paradox of pair programming. *MIS Quarterly, 33*(1), 91.

BASKERVILLE R, LEVINE L, PRIES-HEJE J, RAMESH B and SLAUGHTER S (2002) *Balancing quality and agility in internet speed software development*. Paper presented at the ICIS.

BEAVERS PA (2007) Managing a large" agile" software engineering organization. In *Agile Conference (AGILE), 2007*, pp. 296-303.

BECK K and ANDRES C (2004) *Extreme programming explained: Embrace change*. Addison-Wesley Professional, Upper Saddle River, NJ.

BECK K, et al. (2001) Manifesto for agile software development. http://agilemanifesto.org/, accessed March 1, 2014.

BOEHM B and TURNER R (2004) Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *26th International Conference on Software Engineering (ICSE)*, pp. 718-719.

BOEHM B and TURNER R (2005) Management challenges to implementing agile processes in traditional development organizations. *Software, IEEE, 22*(5), 30-39.

BOWERS AN, SANGWAN RS and NEILL CJ (2007) Adoption of xp practices in the industry—a survey. *Software process: improvement and practice, 12*(3), 283-294.

BROOKS FP (1995) *The mythical man-month (anniversary ed.)*. Addison-Wesley Longman, Boston, MA.

COCKBURN A and HIGHSMITH J (2001) Agile software development, the people factor. *Computer, 34*(11), 131-133.

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

CONBOY K, COYLE S, XIAOFENG W and PIKKARAINEN M (2011) People over process: Key challenges in agile development. *IEEE Software, 28*(4), 48-57.

CONBOY K and FITZGERALD B (2004) *Toward a conceptual framework of agile methods: A study of agility in different disciplines*. Paper presented at the ACM Workshop on Interdisciplinary Software Engineering Research, Newport Beach, CA.

CONBOY K and FITZGERALD B (2010) Method and developer characteristics for effective agile method tailoring: A study of xp expert opinion. *ACM Transactions on Software Engineering and Methodology, 20*(1), 2-30.

DELONE WH and MCLEAN ER (1992) Information systems success: The quest for the dependent variable. *Information systems research, 3*(1), 60-95.

DIBBERN J, WINKLER J and HEINZL A (2008) Explaining variations in client extra costs between software projects offshored to India. *MIS quarterly*, 333-366.

DINGSØYR T, NERUR S, BALIJEPALLY V and MOE NB (2012) A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213-1221.

DYBÅ T (2002) Enabling software process improvement: An investigation of the importance of organizational issues. *Empirical Software Engineering, 7*(4), 387-390.

EISENHARDT KM (1989) Building theories from case study research. *Academy of Management Review, 14*(4), 532-550.

ERICKSON J, LYYTINEN K and SIAU K (2005) Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management, 16*(4), 88-100.

FITZGERALD B, HARTNETT G and CONBOY K (2006) Customising agile methods to software practices at intel shannon. *European Journal of Information Systems, 15*(2), 200-213.

FITZGERALD B, RUSSO NL and STOLTERMAN E (2002) *Information systems development: Methods in action*. McGraw-Hill Education, New York, NY.

GLASER BG (1978) *Theoretical sensitivity: Advances in the methodology of grounded theory*. Sociology Press, Mill Valley, CA.

HARTWICK J and BARKI H (1994) Explaining the role of user participation in information system use. *Management science, 40*(4), 440-465.

HELQUISET JH, DEOKAR A, MESERVY T and KRUSE J (2011) Dynamic collaboration: Participant-driven agile processes for complex tasks. *Data Base for Advances in Information Systems, 42*(2), 95-115.

HIGHSMITH J (2010) Beyond scope, schedule, and cost: The agile triangle. http://jimhighsmith.com/beyond-scope-schedule-and-cost-the-agile-triangle/, accessed January 26, 2016, 2016.

HOLLAND JH (1995) *Hidden order: How adaptation builds complexity*. Basic Books, New York, NY.

HUGHES R (2008) *Agile data warehousing: Delivering world-class business intelligence systems using scrum and xp*. IUniverse.

HUMMEL M (2014) State-of-the-art: A systematic literature review on agile information systems development. In *38th Hawaii International Conference on System Sciences (HICSS)*, pp. 4712-4721, Big Island, HI.

IIVARI J and IIVARI N (2011) The relationship between organizational culture and the deployment of agile methods. *Information and software technology, 53*(5), 509.

JAY J (2013) Navigating paradox as a mechanism of change and innovation in hybrid organizations. *Academy of Management Journal, 56*(1), 137-159.

KIM E and RYOO S (2012) *Agile adoption story from nhn*. Paper presented at the Computer Software and Applications Conference Izmir, Turkey.

KIM G, BEHR K and SPAFFORD G (2014) *The phoenix project: A novel about IT, devops, and helping your business win*. IT Revolution, Portland, OR.

KNIBERG H and IVARSSON A (2012) Scaling agile @ spotify. https://dl.dropboxusercontent.com/u/1018963/Articles/SpotifyScaling.pdf, accessed March 10, 2015.

LARMAN C and BASILI VR (2003) Iterative and incremental developments: A brief history. *Computer, 36*(6), 47-56.

LEE G and XIA W (2010) Toward agile: An integrated analysis of quantitative and qualitative field data on software development quality. *MIS Quarterly, 34*(1), 87-114.

LEVINA N and VAAST E (2008) Innovating or doing as told? Status differences and overlapping boundaries in offshore collaboration. *MIS quarterly*, 307-332.

LEWIS MW (2000) Exploring paradox: Toward a more comprehensive guide. *Academy of Management review, 25*(4), 760-776.

LONG K and STARR D (2008) *Agile supports improved culture and quality for healthwise*. Paper presented at the Agile Conference, Toronto, Canada.

LYYTINEN K and ROSE GM (2006) Information system development agility as organizational learning. *European Journal of Information Systems, 15*(2), 183-199.

MANGALARAJ G, MAHAPATRA R and NERUR S (2009) Acceptance of software process innovations - the case of extreme programming. *European Journal of Information Systems, 18*(4), 344-354.

MARKUS ML and MAO J-Y (2004) Participation in development and implementation-updating an old, tired concept for today's IS contexts. *Journal of the Association for Information Systems, 5*(11), 14.

MARUPING LM, VENKATESH V and AGARWAL R (2009) A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research, 20*(3), 377-399.

MATHIASSEN L and VAINIO AM (2007) Dynamic capabilities in small software firms: A sense-and-respond approach. *IEEE Transactions on Engineering Management, 54*(3), 522-538.

MENS T and TOURWÉ T (2004) A survey of software refactoring. *Software Engineering, IEEE Transactions on, 30*(2), 126-139.

MERRIAM-WEBSTER ONLINE (2015) Dictionary and thesaurus - merriam-webster online. http://www.merriam-webster.com/, accessed July 20, 2015, 2015.

MILES MB and HUBERMAN AM (1994) *Qualitative data analysis: An expanded sourcebook*. SAGE Publications, Thousand Oaks, CA.

MURPHY-HILL E, PARNIN C and BLACK AP (2012) How we refactor, and how we know it. *Software Engineering, IEEE Transactions on, 38*(1), 5-18.

OPDYKE WF (1992) *Refactoring object-oriented frameworks.* University of Illinois at Urbana-Champaign.

PIKKARAINEN M, HAIKARA J, SALO O, ABRAHAMSSON P and STILL J (2008) The impact of agile practices on communication in software development. *Empirical Software Engineering, 13*(3), 303-337.

POPPENDIECK M and POPPENDIECK T (2003) *Lean software development: An agile toolkit*. Addison-Wesley Professional.

QUINN RE and CAMERON KS (1988) *Paradox and transformation: Toward a theory of change in organization and management*. Ballinger, Cambridge, MA.

QUMER A and HENDERSON-SELLERS B (2006) Measuring agility and adaptibility of agile methods: A 4 dimensional analytical tool. In *IADIS International Conference Applied Computing*, pp. 503-507, San Sebastian, Spain.

RAMESH B, MOHAN K and CAO L (2012) Ambidexterity in agile distributed development: An empirical investigation. *Information Systems Research, 23*(2), 323-339.

RODRIGUEZ P, MARKKULA J, OIVO M and TURULA K (2012) Survey on agile and lean usage in finnish software industry. In *International Symposium on Empirical Software Engineering and Measurement*, pp. 139-148, Lund, Sweden.

SANCHEZ J, WILLIAMS L and MAXIMILIEN EM (2007) A longitudinal study of the use of a test-driven development practice in industry. In *Proc. Agile*.

SARKER S, MUNSON CL, SARKER S and CHAKRABORTY S (2009) Assessing the relative contribution of the facets of agility to distributed systems development success: An analytic hierarchy process approach. [Article]. *European Journal of Information Systems, 18*(4), 285-299.

SARKER S and SARKER S (2009) Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research, 20*(3), 440-461.

SARKER S, XIAO X and BEAULIEU T (2013) Guest editorial: Qualitative studies in information systems: A critical review and some guiding principles. *MIS Quarterly, 37*(4), iii-xviii.

SCHWABER K and BEEDLE M (2003) *Agile software development with scrum*. Prentice Hall, Upper Saddle River, NJ.

SEDDON PB (1997) A respecification and extension of the delone and mclean model of IS success. *Information systems research, 8*(3), 240-253.

SMITH WK and LEWIS MW (2011) Toward a theory of paradox: A dynamic equilibrium model of organizing. *Academy of Management Review, 36*(2), 381-403.

STRODE DE (2015) A dependency taxonomy for agile software development projects. *Information Systems Frontiers*, 1-24.

STRODE DE, HUFF SL, HOPE B and LINK S (2012) Coordination in co-located agile software development projects. *The Journal of Systems and Software, 85*(6), 1222-1238.

TRUEX D, BASKERVILLE R and TRAVIS J (2000) Amethodical systems development: The deferred meaning of systems development methods. *Accounting, management and information technologies, 10*(1), 53-79.

TURNER R and BOEHM B (2003) People factors in software management: Lessons from comparing agile and plan-driven methods. *CrossTalk Magazine, The Journal of Defense Software Engineering, 16*(12), 4-8.

URQUHART C (2012) *Grounded theory for qualitative research: A practical guide*. SAGE Publications, Thousand Oaks, CA.

VAN DE VEN AH and POOLE MS (1988) Paradoxical requirements for a theory of change. In *Paradox and transformation: Toward a theory of change in organization and management* (QUINN RE and CAMERON KS, Eds), pp. 19-64, Ballinger, Cambridge, MA.

VERSIONONE.COM (2013) Seventh annual state of agile development survey. http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf, accessed June 16, 2014.

VIDGEN R and WANG X (2009) Coevolving systems and the organization of agile software development. *Information Systems Research, 20*(3), 355-376.

WEIYIN H, THONG JYL, CHASALOW LC and DHILLON G (2011) User acceptance of agile information systems: A model and empirical test. [Article]. *Journal of Management Information Systems, 28*(1), 235-272.

WUFKA M (2013) *Domain understanding in agile information systems development: The role of conceptual modeling.* PhD thesis, University of British Columbia, Vancouver, Canada.

YIN RK (2013) *Case study research: Design and methods*. Sage publications, Thousand Oaks, CA.

Appendix 3.1: Interview Protocol

This interview guide is separated in different sections. It is used and adapted for different categories of stakeholders such as developers and analysts, managers, user representatives and end users, and executives who initiated the project. As we use semi-structured interviews, we follow users' recollections triggered by our questions while ensuring that we get answers to as many questions as possible without forcing respondents to answer the questions in order.

Note: This interview guide was initially written in French. For the sake of consistency with the rest of this document, it has been translated into English by the author.

Introduction

1. Although I may be familiar with some of the terms you may mention, please take the time to explain them to me as if they were unbeknownst to me.
2. If at any point in time my questions seem unclear or irrelevant, please do not hesitate to interrupt me to get more details
3. In this guide, the use of the masculine pronoun is used for brevity purposes and to simplify the formulation of questions
4. Terminology
    a. In this study, I refer to "projects" as undertakings where a software artifact is being developed for customers, whether they be internal or external. This software may be developed by internal staff only or a mix of internal staff and external resources such as consultants, be they part of a firm or self-employed
5. This study is conducted in the context of a PhD in administration at HEC Montreal. It deals with the concept of agility in the context of IT projects involving the development of software artifacts, commonly referred to as information systems development (ISD) projects.

Context of the study

The main objective of this research is to understand how work practices and the use of tools supporting those practices may promote or reflect agility in ISD projects. At the same time, we seek to explore certain contextual or structural factors which may enable or hinder the application of the concept of agility in a project. Finally, we would like to see if there are any specific changes which may be associated with these practices when (and if) they become adopted for more than one project.

In order to fulfill this objective, I need to get a good understanding of your work practices and your perspective as to how the project unfolded over time. As I gather different stakeholders' perspectives on a given project, I am able to reconstruct a more comprehensive timeline and overall depiction of this project. The work practices we will discuss in this interview relate to your everyday job, more specifically those enacted during the course of the project and their potential differences with other practices which are typically recommended or prescribed for other projects. However, they may also deal with other work practices which are usually associated with activities located outside of the project's direct sphere of influence (e.g., human resources, budgeting etc.).

Confidentiality

1. This study has been approved by Comité d'Ethique à la Recherche de l'école des Hautes Etudes Commerciales (HEC) de Montréal. The confidentiality of the information we exchange is guaranteed by a content form which we will both sign. Please take the time to review this form and ask me any questions you may have about it before signing it.
2. Recording
    a. [Take out recorder]
    b. If you agree to have me record this interview, you may decide at any point in time to stop, momentarily or permanently, the recording. If so, please just let me know.
3. Process
    a. During this interview, I will ask you a series of questions. If you would like to add any other information or refer to other sources of documentation or information to complete your answers please do not hesitate to let me know.
4. Contacts
    a. If you would like to refer to other people who may be interested in participating in this study or who may provide me with relevant information on this study, please do not hesitate to let me know.
5. [Start recording if respondent agreed to recording]

Appendix 3.2: Interview Guide

Introduction

- Can you provide me with your name, your employer's name and your job description(s)?
- How long have you been employed at this company?
- Can you briefly provide me with some background information regarding your education and job history?

In this section, we will discuss the practices in place in the organization(s) involved in the project.

- How are projects of this scale and type typically organized (kickoff, budget etc.)?
- What type of constraints to they usually face? How do these compare for the project we are discussing in detail today?
- Is there a specific ISD method which is typically followed?
- Is there a specific project management method which is typically followed?
- Are there any specific tools which are typically used in projects?
- To what extent would you say that are you typically free to organize your work and the practices you follow?
- How do you determine, within your team, which practices to follow?

In this section, we will discuss the project in which you took/are taking part.

- Can you briefly describe the project and its objectives?
- According to you, what are the factors which influenced the undertaking of this project at this point in time for the organization?
- Can you provide me with a brief timeline of the project?
- When did you get involved in the project? Until when were you involved?
- What was your role in this project? Did it change over time?
- Have you been involved with any other projects in this organization? If so, which one(s)?
- Were there any important moments which have marked this project's unfolding? Which one(s)? For example, you may wish to think of 3 positive moments and 3 negative moments.
- In your opinion, what triggered those moments?

- How did you feel throughout this project?
- How do think that other team members felt during this project? Did you discuss these feeling with one another?
- How did you follow the project's progress?
- How did you follow the project's success?

In this section, we will discuss the practices used to conduct the project in which you took part.

- How would you describe the approach followed to undertake this project? Was this a statement made by particular project stakeholder is this your reflective opinion?
- How did the team decide which practices to adopt for the duration of the project?
- Where there any prescriptions, be they using tools or practices, which influenced the decision to use certain tools or enact certain practices?
- Did the practices enacted for the project create tensions within the team or threaten the project's completion at any point in time? If so how were these addressed?
- How much freedom did you have to select the tools which you thought were most appropriate to tackle the project? Were any tools forced upon the team inappropriate for the project in your opinion?
- Did the requirements of the project change during its course? If so, how was this handled?
- How involved were you with discussing issues and features with users during the project?
- Were there any dependencies across projects which needed to be resolved in order to allow this project to move forward

Conclusion

- Do you think that following this project, some of the practices or the approach enacted will "stick" in the organization if they are new?
- Why do you think that agility was required for this project?
- In what ways could the concept agility have been applied further to help the project, if possible?
- What would say is your definition of agility in ISD projects?

- Are there any other pieces of information or anecdotes you may think of and I have not asked about which you would like to mention before we conclude this interview?
- Thank you for your time.

Appendix 3.3: Structuring Chains of Evidence in a Relational Model

In the model presented in Figure A3.3.1, each case is represented by an entity. This entity holds all of the case's chains of evidence. This data was automatically imported into the database from the spreadsheet where they were created during the first round of coding. The original coded quotes were also imported from the MaxQDA project in each case's quote entity. A many-to-many entity links chains of evidence to quotes, allowing quotes to be linked to multiple chains of evidence. For example, the MarketBI entity holds chains of evidence for the MarketBI case while MarketBI_Quote holds the coded quotes for MarketBI. X_MarketBI_MarketBI_Quote provides a link between MarketBI's chains of evidence and MarketBI_Quote's coded quotes. The same pattern was applied for all five cases.

To provide access to the chains of evidence from all cases, a view, AllCaseView, was created. It contains the same attributes as the entities for chains of evidence and also includes a Case_ID attribute indicating to which case a given chain of evidence pertains. The implementation of AllCaseView is a simple SELECT statement against each case's main entity appended to one another using a UNION operation.

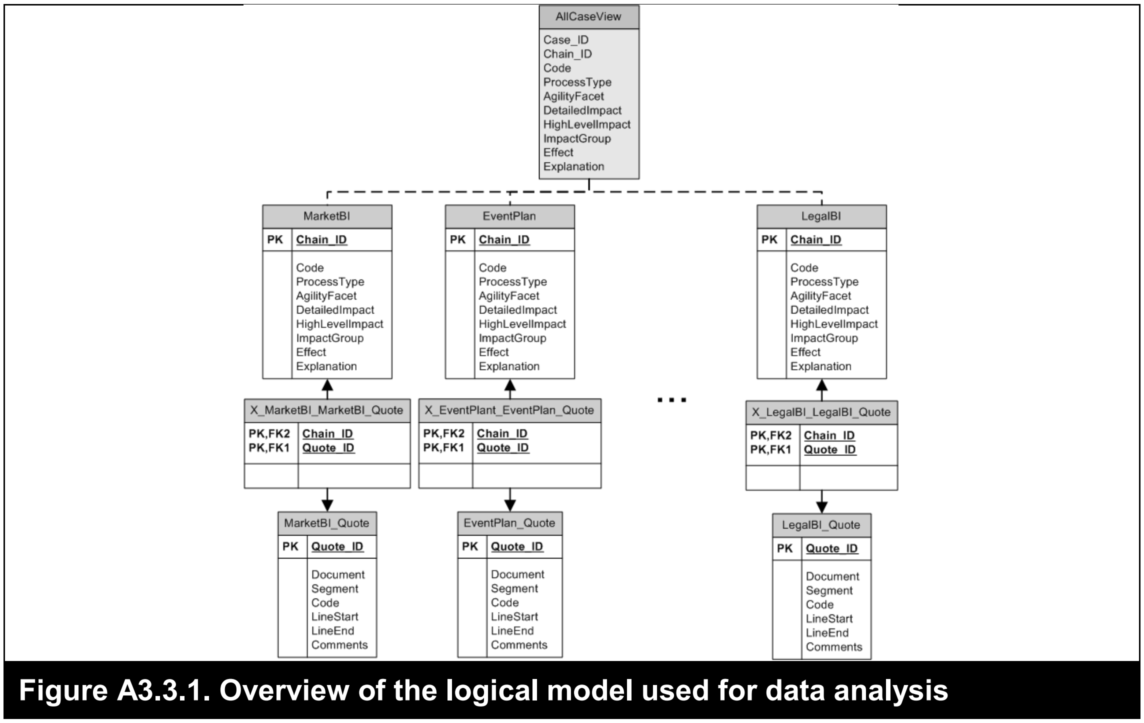
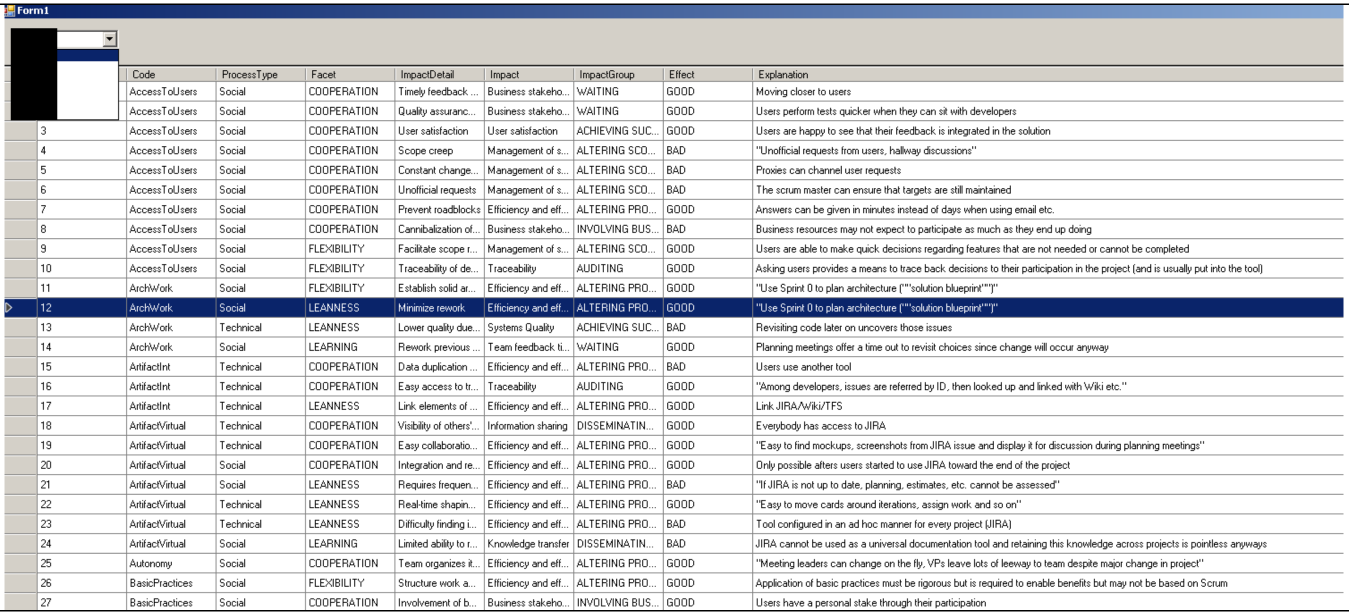


Figure A3.3.1. Overview of the logical model used for data analysis

Appendix 3.4: Designing Artifacts to Enforce the Traceability of Evidence during Data Analysis

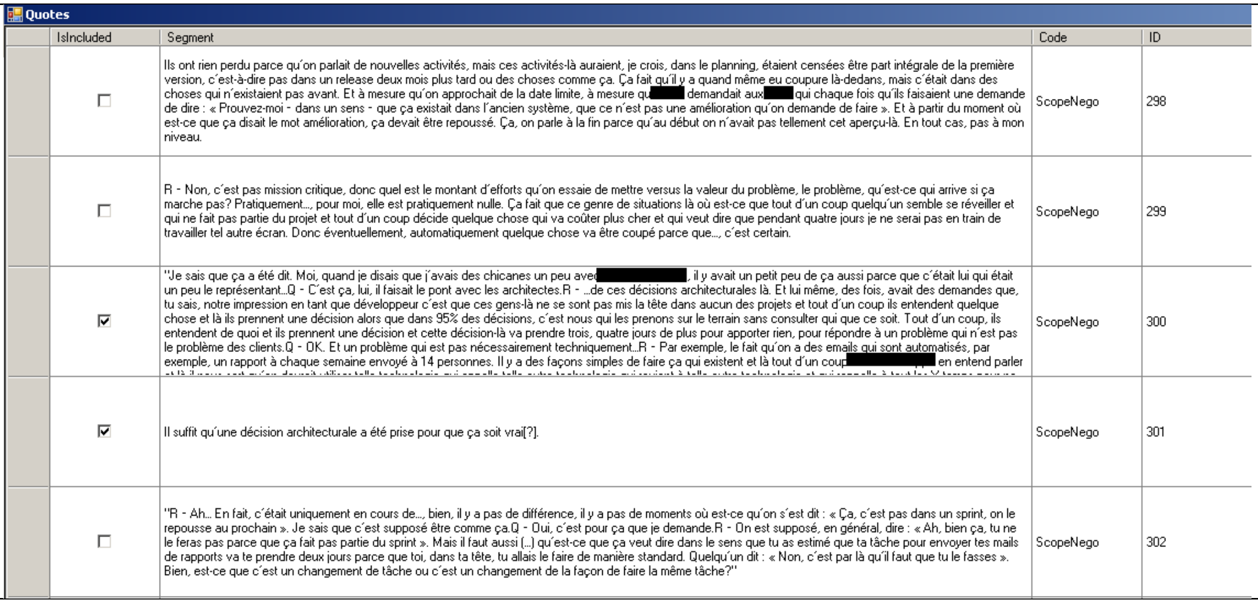
The artifact designed to modify the association between a coded segment and a chain of evidence was built to (1) provide an intuitive user interface to visualize data and (2) easily add or remove links between coded segments and chains of evidence. Indeed, the modification of those links by hand involves issuing a number of queries against the relational database and is time consuming and error prone. The artifact was designed to provide a layer of abstraction to encourage, rather than discourage the reviewing of those links. Figure A3.4.1 shows a screenshot of the main form where chains of evidence are loaded from the database and displayed upon selecting a case in the combo box in the upper left corner. Figure A3.4.2 shows the coded quotes for the case, along with a checked checkbox for quotes that are currently associated with the chain of evidence selected in the main form. The user can add or remove links simply by checking or unchecking the checkbox next to a quote.



	Code	ProcessType	Facet	ImpactDetail	Impact	ImpactGroup	Effect	Explanation
[redacted]	AccessToUsers	Social	COOPERATION	Timely feedback ...	Business stakeho...	WAITING	GOOD	Moving closer to users
[redacted]	AccessToUsers	Social	COOPERATION	Quality assuranc...	Business stakeho...	WAITING	GOOD	Users perform tests quicker when they can sit with developers
3	AccessToUsers	Social	COOPERATION	User satisfaction	User satisfaction	ACHIEVING SUC...	GOOD	Users are happy to see that their feedback is integrated in the solution
4	AccessToUsers	Social	COOPERATION	Scope creep	Management of s...	ALTERING SCO...	BAD	"Unofficial requests from users, hallway discussions"
5	AccessToUsers	Social	COOPERATION	Constant change...	Management of s...	ALTERING SCO...	BAD	Proxies can channel user requests
6	AccessToUsers	Social	COOPERATION	Unofficial requests	Management of s...	ALTERING SCO...	BAD	The scrum master can ensure that targets are still maintained
7	AccessToUsers	Social	COOPERATION	Prevent roadblocks	Efficiency and eff...	ALTERING PRO...	GOOD	Answers can be given in minutes instead of days when using email etc.
8	AccessToUsers	Social	COOPERATION	Cannibalization of...	Business stakeho...	INVOLVING BUS...	BAD	Business resources may not expect to participate as much as they end up doing
9	AccessToUsers	Social	FLEXIBILITY	Facilitate scope r...	Management of s...	ALTERING SCO...	GOOD	Users are able to make quick decisions regarding features that are not needed or cannot be completed
10	AccessToUsers	Social	FLEXIBILITY	Traceability of de...	Traceability	AUDITING	GOOD	Asking users provides a means to trace back decisions to their participation in the project (and is usually put into the tool)
11	ArchWork	Social	FLEXIBILITY	Establish solid ar...	Efficiency and eff...	ALTERING PRO...	GOOD	"Use Sprint 0 to plan architecture (""solution blueprint"")"
12	ArchWork	Social	LEANNESS	Minimize rework	Efficiency and eff...	ALTERING PRO...	GOOD	"Use Sprint 0 to plan architecture (""solution blueprint"")"
13	ArchWork	Technical	LEANNESS	Lower quality due...	Systems Quality	ACHIEVING SUC...	BAD	Revisiting code later on uncovers those issues
14	ArchWork	Social	LEARNING	Rework previous ...	Team feedback ti...	WAITING	GOOD	Planning meetings offer a time out to revisit choices since change will occur anyway
15	ArtifactInt	Technical	COOPERATION	Data duplication ...	Efficiency and eff...	ALTERING PRO...	BAD	Users use another tool
16	ArtifactInt	Technical	COOPERATION	Easy access to tr...	Traceability	AUDITING	GOOD	"Among developers, issues are referred by ID, then looked up and linked with Wiki etc."
17	ArtifactInt	Technical	LEANNESS	Link elements of ...	Efficiency and eff...	ALTERING PRO...	GOOD	Link JIRA/Wiki/TFS
18	ArtifactVirtual	Technical	COOPERATION	Visibility of others'...	Information sharing	DISSEMINATIN...	GOOD	Everybody has access to JIRA
19	ArtifactVirtual	Technical	COOPERATION	Easy collaboratio...	Efficiency and eff...	ALTERING PRO...	GOOD	"Easy to find mockups, screenshots from JIRA issue and display it for discussion during planning meetings"
20	ArtifactVirtual	Social	COOPERATION	Integration and re...	Efficiency and eff...	ALTERING PRO...	GOOD	Only possible afters users started to use JIRA toward the end of the project
21	ArtifactVirtual	Social	LEANNESS	Requires frequen...	Efficiency and eff...	ALTERING PRO...	BAD	"If JIRA is not up to date, planning, estimates, etc. cannot be assessed"
22	ArtifactVirtual	Technical	LEANNESS	Real-time shapin...	Efficiency and eff...	ALTERING PRO...	GOOD	"Easy to move cards around iterations, assign work and so on"
23	ArtifactVirtual	Technical	LEANNESS	Difficulty finding i...	Efficiency and eff...	ALTERING PRO...	BAD	Tool configured in an ad hoc manner for every project (JIRA)
24	ArtifactVirtual	Social	LEARNING	Limited ability to r...	Knowledge transfer	DISSEMINATIN...	BAD	JIRA cannot be used as a universal documentation tool and retaining this knowledge across projects is pointless anyways
25	Autonomy	Social	COOPERATION	Team organizes it...	Efficiency and eff...	ALTERING PRO...	GOOD	"Meeting leaders can change on the fly, VPs leave lots of leeway to team despite major change in project"
26	BasicPractices	Social	FLEXIBILITY	Structure work a...	Efficiency and eff...	ALTERING PRO...	GOOD	Application of basic practices must be rigorous but is required to enable benefits but may not be based on Scrum
27	BasicPractices	Social	COOPERATION	Involvement of b...	Business stakeho...	INVOLVING BUS...	GOOD	Users have a personal stake through their participation

Note : black squares have been added to hide confidential information

Figure A3.4.1. Main screen of coding artifact – project selection and displaying chains of evidence



	IsIncluded	Segment	Code	ID
	□	Ils ont rien perdu parce qu´on parlait de nouvelles activités, mais ces activités-là auraient, je crois, dans le planning, étaient censées être part intégrale de la première version, c´est-à-dire pas dans un release deux mois plus tard ou des choses comme ça. Ça fait qu´il y a quand même eu coupure là-dedans, mais c´était dans des choses qui n´existaient pas avant. Et à mesure qu´on approchait de la date limite, à mesure qu[redacted] demandait aux[redacted] qui chaque fois qu´ils faisaient une demande de dire : « Prouvez-moi - dans un sens - que ça existait dans l´ancien système, que ce n´est pas une amélioration qu´on demande de faire ». Et à partir du moment où est-ce que ça disait le mot amélioration, ça devait être repoussé. Ça, on parle à la fin parce qu´au début on n´avait pas tellement cet aperçu-là. En tout cas, pas à mon niveau.	ScopeNego	298
	□	R - Non, c´est pas mission critique, donc quel est le montant d´efforts qu´on essaie de mettre versus la valeur du problème, le problème, qu´est-ce qui arrive si ça marche pas? Pratiquement..., pour moi, elle est pratiquement nulle. Ça fait que ce genre de situations là où est-ce que tout d´un coup quelqu´un semble se réveiller et qui ne fait pas partie du projet et tout d´un coup décide quelque chose qui va coûter plus cher et qui veut dire que pendant quatre jours je ne serai pas en train de travailler tel autre écran. Donc éventuellement, automatiquement quelque chose va être coupé parce que..., c´est certain.	ScopeNego	299
	☑	"Je sais que ça a été dit. Moi, quand je disais que j´avais des chicanes un peu ave[redacted], il y avait un petit peu de ça aussi parce que c´était lui qui était un peu le représentant...Q - C´est ça, lui, il faisait le pont avec les architectes.R - ...de ces décisions architecturales là. Et lui même, des fois, avait des demandes que, tu sais, notre impression en tant que développeur c´est que ces gens-là ne se sont pas mis la tête dans aucun des projets et tout d´un coup ils entendent quelque chose et là ils prennent une décision alors que dans 95% des décisions, c´est nous qui les prenons sur le terrain sans consulter qui que ce soit. Tout d´un coup, ils entendent de quoi et ils prennent une décision et cette décision-là va prendre trois, quatre jours de plus pour apporter rien, pour répondre à un problème qui n´est pas le problème des clients.Q - OK. Et un problème qui est pas nécessairement techniquement...R - Par exemple, le fait qu´on a des emails qui sont automatisés, par exemple, un rapport à chaque semaine envoyé à 14 personnes. Il y a des façons simples de faire ça qui existent et là tout d´un coup[redacted] en entend parler	ScopeNego	300
	☑	Il suffit qu´une décision architecturale a été prise pour que ça soit vrai[?].	ScopeNego	301
	□	"R - Ah... En fait, c´était uniquement en cours de..., bien, il y a pas de différence, il y a pas de moments où est-ce qu´on s´est dit : « Ça, c´est pas dans un sprint, on le repousse au prochain ». Je sais que c´est supposé être comme ça.Q - Oui, c´est pour ça que je demande.R - On est supposé, en général, dire : « Ah, bien ça, tu ne le feras pas parce que ça fait pas partie du sprint ». Mais il faut aussi (...) qu´est-ce que ça veut dire dans le sens que tu as estimé que ta tâche pour envoyer tes mails de rapports va te prendre deux jours parce que toi, dans ta tête, tu allais le faire de manière standard. Quelqu´un dit : « Non, c´est par là qu´il faut que tu le fasses ». Bien, est-ce que c´est un changement de tâche ou c´est un changement de la façon de faire la même tâche?"	ScopeNego	302

Note : black squares have been added to hide confidential information

Figure A3.4.2. Linking screen of coding artifact – Displaying coded segments for a selected chain of evidence from the main screen

Appendix 3.5: Alternating Perspectives on Chains of Evidence for Data Analysis

One of the issues we have experienced with spreadsheets and relational data during data analysis is their tabular format. Indeed, these technologies have a propensity to encourage the breaking down break down the data into separate attributes or columns. While this is foundation of the operations one can perform against relational data, it provides a narrow perspective on that data. For instance, studying chains of evidence for a particular column (e.g., process type) across cases is done easily. However, this process discourage a syntactic, row-based view of the data where the columns are joined together to form a sentence. To help provide this additional perspective on our data, we built a series of queries that take chains of evidence and converts them into sentences in plain English. While this does not in itself reveal any new pattern in the data, it provides the researcher with a fresh perspective on that data and can help suggest patterns that were not found before. Table A3.5.1 shows how a quote becomes part of a chain of evidence and how this chain of evidence can later be converted into plain English to assist with its analysis.

Table A3.5.1. Rebuilding standard sentences from chains of evidence

Original quote	“Bien oui, bien oui. Et si le client…, et encore là, peut-être qu’il aurait été ouvert à ça, j’ai juste pas pris la chance. Si le client comprend que c’est un apprentissage pour lui, d’un point de vue global, et qu’il va être gagnant à long terme, bien, je peux pas le communiquer, mais moi il faut que je le fasse. Il y a des choses que je dois laisser aller, la loi de Murphy va faire sa job toute seule. S’il y a quelque chose qui faut qui aille mal, ça va aller mal et après ça on va apprendre de ça et on va se réaligner. Ça fait que j’y vais comme ça” [Project manager, MarketBI]
Chain of evidence	• Code: Not TransparentComm • ProcessType: Social • AgilityFacet: Learning • DetailedImpact: Give time and space for the team to learn • HighLevelImpact: Efficiency and effectiveness • ImpactGroup: Altering Productivity • Effect: GOOD • Explanation: Temporary added costs to enable team learning and ensure that peer review process becomes firmly anchored in the team’s regular practices
Reconstituted sentence from chain of evidence (via SQL query)	A social process, Not transparentComm, deals with learning and impacts altering productivity - efficiency and effectiveness (give time and space for the team to learn) in a good way because of temporary added costs to enable team learning and ensure that

	peer review process becomes firmly anchored in the team’s regular practices.

Chapter 4: Essay 3 – Conceptualizing IS Development as an Organizational Routine: Implications and Avenues for Research

Abstract

Despite the abundance of frameworks and methods available to structure the information systems development process (ISD), it remains a complex and uncertain endeavor. Literature on the topic has traditionally approached ISD from one of two perspectives: (1) as a defined process that can be reproduced through the use of methods providing series of prescriptions; and (2) as an emergent process that relies on the improvisation of actors to cope with unforeseen contingencies. Arguing that ISD encompasses both these aspects, we propose a conceptualization of ISD as a series of activities based on the software development life cycle that builds on Feldman and Pentland's ontology of organizational routines. As an organizational phenomenon, we propose that ISD includes two mutually constitutive aspects: (1) an ostensive aspect, the generalized idea as to how ISD should unfold in principle and (2) a performative aspect that represents how ISD unfolds in practice based on the agency of the actors taking part in its enactment. We further acknowledge the role of artifacts as carriers of the ostensive aspect of ISD as well as enabling and constraining elements of performances. To illustrate the potential of this conceptualization, we turn to the literature on Agile ISD and explain how enduring issues on the topic may be addressed by this conceptualization. Seeking to foster research on Agile ISD that can help address those issues, we develop six avenues for future research using our conceptualization of ISD as an organizational routine. These avenues focus on the ostensive and performative aspects of Agile ISD, the role of artifacts in the Agile ISD process as well as the dyadic interactions that exist between these three components.

Keywords

Information systems development, agile ISD, agile software development, organizational routines, performative aspect, ostensive aspect, software development lifecycle.

4.1. Introduction

Information systems development (ISD) is a complex (e.g., Avison & Fitzgerald, 2006; Hirschheim & Klein, 1989) and uncertain (e.g., Ang & Beath, 1993; Harris et al., 2009) endeavor. Within this context, practitioners rely on methods or frameworks to guide the ISD process. Notwithstanding, it has been noted that those methods are often tailored to fit the organization and the project where they are used, shaping the performances of organizational actors over time (Iivari & Maansaari, 1998). ISD therefore encompasses methods as artifacts providing patterns of prescribed behaviors as well as series of "practices involved in the process of designing, building, implementing, and maintaining an information system" (Conboy, 2009:329). In line with these two views, the literature offers two overarching perspectives on ISD and we argue that the study of the relationship between these two perspectives has the potential to provide new insight on the topic.

The *mechanistic perspective* conceptualizes ISD as a process made of a series of operations that are well-defined and carried out by actors who possess the business and technical expertise to fulfill them according to expectations (Brinkkemper, 1996). This perspective is primarily concerned with the elaboration and application of *methods-as-theories* designed to accomplish the ISD process in the most optimal manner and transpose it in other contexts. Conversely, the *emergent perspective* acknowledges the non-routine aspect of ISD and the need for actors to constantly improvise in their *practice* of ISD (Orlikowski, 1993). While expectations may be formed regarding the process of ISD, its actual unfolding is influenced by the complexity and uncertainty associated with the building of a virtual artifact.

While the emergent perspective has enjoyed popularity in recent years through its ability to account for variations and contextual specificities that influence the performance of ISD, we argue, in line with the mechanistic perspective, that there is a stable component to the ISD process and its enactment over time that must also be accounted for in studies on ISD. In addition to this stable component, we acknowledge the importance of the *performance* of ISD in line with the emergent perspective. Indeed, it has been shown that over time, performances can alter the stable pattern of actions outlined by methods (e.g., Fitzgerald et al., 2006; Sarker & Sarker, 2009). Our work offers to study the relationship between these two perspectives as a means to better understanding the relevance of ISD methods as theories performed by organizational actors in practice.

To account for these two aspects of ISD as well as their relationship, we propose a conceptualization of ISD based on Feldman and Pentland's (2003) ontology of organizational routines. Organizational routines are defined as “repetitive, recognizable, patterns of interdependent actions, carried out by multiple actors” (Feldman & Pentland, 2003:95) and are mutually constituted by two aspects. The *ostensive aspect* of a routine is the shared perception of the routine which may remain stable over time. The *performative* aspect of a routine is the actual enactment of actions for which actors may exert agency and choose from a “repertoire of possibilities” (Feldman & Pentland, 2003:102). Within this ontology, artifacts play an important role as tangible references that help diffuse the ostensive aspect a routine while enabling and constraining the performances of actors (Pentland & Feldman, 2005).

Applied to the context of ISD, this ontology accounts for the shared understanding that ISD is a process carried over time that involves a series of generic actions (e.g., formulating requirements) while accounting for the ability for actors to improvise in its performance. The mutually constitutive aspects of organizational routines provide a theoretical foundation to explain how the ostensive aspect provides structure for the enactment of performances while performances trigger the evolution of the ostensive aspect over time. The numerous software tools and documentation that actors refer to in their performance of ISD are part of the artifacts that support these two aspects.

To illustrate the potential of this conceptualization, we propose its application to the context of Agile ISD. Building on past research on the topic (Conboy, 2009; Dybå & Dingsøyr, 2008; Hummel, 2014), we highlight five enduring issues which we categorize based on their epistemological or phenomenological nature and explain how a conceptualization of ISD as an organizational routine may, in the context of Agile ISD, help to address those issues.

We then build on the work of Pentland and Feldman (2005) to propose six main avenues of research on Agile ISD. These avenues focus on: (1) the study of performances and the influence of the context in their enactment and their implications at multiple levels; (2) the ostensive aspect and the ability for multiple ostensive aspects to coexist across groups of actors; (3) artifacts and their role as illustrations of the ostensive and performative aspects of a routine; (4) the interaction between the ostensive and performative aspects of ISD; (5) interactions between performances and artifacts, more specifically the enabling and constraining power that artifacts have on the enactment of performances; and (6) interactions between artifacts and the ostensive aspect of ISD, such as the ability for artifacts to separate or reconcile multiple, coexisting ostensive aspects. Overall, the novel conceptualization of ISD offered in this work has the potential to yield insight and foster future research on this complex and uncertain phenomenon.

In the next sections, we develop our conceptualization of ISD as an organizational routine. Then we outline the features of the ontology of organizational routines proposed by Feldman and Pentland (2003) and detail their implications for our conceptualization of the ISD process. We illustrate the relevance of our conceptualization of ISD as an organizational routine in the context of Agile ISD and highlight its potential to address some enduring issues faced by research on the topic. Finally we develop avenues for research on Agile ISD showcasing the potential of our conceptualization of ISD and provide some concluding remarks.

4.2. ISD as an Organizational Routine

As part of organizational life, ISD is often considered as a process consisting in a number of generic activities that are carried out regardless of any method of framework in use in an organization (Avison & Fitzgerald, 2006). At the same time, it is acknowledged as a type of complex (Davis, 1982; Harris et al., 2009), non-routine (Maruping et al., 2009; Moe et al., 2012) knowledge work involving highly skilled individuals who wrestle with ever changing requirements and fast-paced technological innovation. Accordingly, researchers have traditionally approached this phenomenon from one of two main perspectives based on their objectives.

The *mechanistic perspective* focuses on the generic character of ISD. While there may be contingencies that emerge during the course of a project, these contingencies can be accounted for either at the onset of the project or at the time of their occurrence. According to the mechanistic perspective, ISD is a *defined process* (Schwaber & Beedle, 2003) that can be structured through prescriptions and guidelines along with series of roles assigned to organizational actors. The formulation of those prescriptions and guidelines is often the result of the scientific elaboration and application of methods as theories that are *engineered* for the purpose of guiding the ISD process (Brinkkemper, 1996). A primary objective of those theories is to provide generalizable prescriptions that can be applied across projects so long as their core assumptions are respected. Traditionally, these methods have focused on structuring the ISD process through the elaboration of a *plan* that dictates the sequence and execution of the work (e.g., Royce, 1970). More recently, advocates of Agile ISD methods (e.g., Beck & Andres, 2004; Schwaber & Beedle, 2003) have acknowledged the need to enact "artful planning" (Baskerville, 2006) and manage uncertainty and complexity as they become relevant throughout the course of a project rather than at its onset. Although it may at first appear that Agile ISD methods are not mechanistic, their prescriptions and rituals (Beck & Andres, 2004; Schwaber & Beedle, 2003) suggest otherwise. Overall, the mechanistic perspective focuses on ISD as a stable, generic pattern of actions which faithful enactment can yield positive outcomes (Beck & Andres, 2004).

The *emergent perspective* acknowledges the non-routine aspect of ISD (Orlikowski, 1993) as part of the organizational processes characterized by emergence and uncertainty. Contrary to the engineering perspective where generalizability is an important outcome of the research process, the practical perspective highlights the relevance of specific patterns of actions under specific conditions. As a result, many works adopting this perspective build *explanations* that help "promote greater understanding or insights by others into the phenomenon of interest" (Gregor, 2006:619). For instance, recent works on Agile ISD have highlighted the relevance of specific coordination mechanisms that enable fast communication between team members, although these mechanisms and the artifacts they rely on differ whether the project is conducted in a small co-located team or a large, distributed workforce (e.g., Sarker & Sarker, 2009). However, the links that exist between the definition of a given method and the enactment of the ISD process by organizational actors are still unexplored, save in instances such as those where contingencies trigger localized customization of those methods (e.g., Cao et al., 2009; Fitzgerald et al., 2002).

While these mechanisms may be promoted by ISD methods, the links that exist between methods and the performance of ISD are largely missing. Indeed, the emergent perspective subscribes the long standing idea that "there is no silver bullet", or "single development, in either technology or management technique, which by itself promises even one order of magnitude improvement in productivity, in reliability, in simplicity"

(Brooks, 1987:11) and puts the method as part of the context without theorizing on its influence toward the performance of ISD.

We posit that taken together, the features of the mechanistic and emergent perspectives provide a means to account for (1) the stability of ISD as a generic organizational process enacted over time (Erickson et al., 2005) and (2) the improvisational nature of complex knowledge work (Faraj & Sproull, 2000; Orlikowski, 1993) as well as (3) their interrelationship. Indeed, ISD methods have been studied as frames that structure the development process while their enactment results in the customization and tailoring of those methods to fit the specificities of a given context (e.g., Fitzgerald et al., 2006). However, the adoption of an ISD method is rarely conceptualized as an evolutionary process that has the potential to fundamentally alter the structures underlying the performance of ISD. Overall, we argue that these two facets of the ISD phenomenon have the potential to reveal insights regarding the emergent performances of ISD within the boundaries a stable frame of reference that actors share, whether it exists in the form of a method or a recurring *history* of developing IS in a particular context. Theoretically speaking, we argue that these two facets call for the use of a perspective which places equal emphasis on both while acknowledging the importance of their relationship.

To that effect, we propose a conceptualization of ISD based on Feldman and Pentland's (2003) ontology of organizational routines. Organizational routines are defined as "repetitive, recognizable, patterns of interdependent actions, carried out by multiple actors," (Feldman & Pentland, 2003:95). We consider ISD as an organizational phenomenon and accordingly conceptualize it as a *business process* based on a number of definitions found in extant literature and presented in Table 4.1.

Table 4.1. Extant definitions of business processes

Definition	Source
"A structured, measured set of activities designed to produce a specific output for a particular customer or market"	Davenport (2013)
"A collection of activities that takes one or more kinds of input and creates an output that is of value to the customer"	Hammer and Champy (2009)
"Series of steps designed to produce a product or service"	Rummler and Brache (2012)
"A systematic series of activities directed towards causing an end result such that one or more inputs will be acted upon to create one or more outputs"	Project Management Institute (2013)

In line with the ontology proposed by Feldmand and Pentland, these definitions highlight the relevance of series of activities involving multiple, interdependent actors and which use a series of inputs to produce a desired output. These activities are defined at varying levels of detail (e.g., assembling components) based on the degree of precision required to understand the value created by their performance. The interdependencies that exist across activities and actors define a general sequence according to which an output can be produced. If a mistake is made at an earlier step, it can impact the remainder of the execution and possibly the output itself. Similarly, some activities must be performed sequentially and offer no possibility for reprocessing in the case of an error. For example, if parts of a computer board are soldered in an incorrect manner, the whole board may need to be scrapped. These linkages not only define the state of an input's transformation at a given point in time but also account for the relationships between the actions performed by the actors involved in the execution of the process.

Similarly, the IS development process involves the creation of a software artifact by multiple actors (e.g., developers, business analysts) using a variety of inputs, some of which may be intangible (e.g., requirements voiced by users). Failures in earlier stages of the process also have an impact on the output produced. For example, poor requirements engineering has been found as a source of failure in ISD projects (e.g., Montealegre & Keil, 2000). In many instances, the definition of the process is based on the use of an ISD method that structures work based on a series of pre-defined activities. Indeed, an ISD method provides a "coherent and systematic approach, based on a particular philosophy of systems development, which will guide developers on what steps to take, how these steps should be performed, and why these steps are important in the development of an information system" (Fitzgerald et al., 2002:5).

Notwithstanding the relevance of methods in the context of ISD, a definition of the ISD process based solely on the prescriptions of a given method is problematic. Indeed, it restricts the ability for researchers to conceptualize ISD in a variety of contexts where methods differ or may not be used at all, as is often the case (Avison & Fitzgerald, 2006) and calls have been made to offer perspectives to avoid studying ISD exclusively through the prism of methods (Truex et al., 2000). The argument is that methods often focus on specific parts of the ISD process based on their stated purpose. For example, eXtreme Programming (XP, Beck & Andres, 2004) focuses on organizing the work of developers while Scrum (Schwaber & Beedle, 2003) prescribes practices to manage ISD projects.

XP and Scrum both deal with the ISD process but diverge greatly regarding their prescriptions. This renders comparisons across settings where methods differ difficult. In order to provide a point of reference for all these different instances, we propose that the SDLC – as presented in Figure 4.1 – be used as a generic depiction of the ISD organizational routine. Indeed, the SDLC predates many ISD methods and describes a comprehensive array of practices organized in a series of generic activities which extend beyond the coding of an IS (e.g., project planning). This conceptualization has become so central to the way we approach ISD that it is an integral part of IS curricula, as evidenced by the layouts of many IS manuals and textbooks that are structured around the definition and undertaking of those four generic activities (e.g., Avison & Fitzgerald, 2006; Dennis et al., 2008; Valacich et al., 2004).

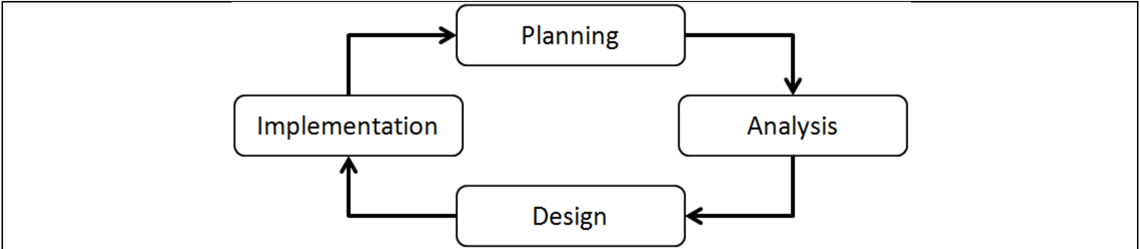


Figure 4.1. The software development life cycle and its main activities (SDLC)

The first activity of the SDLC, *planning*, provides a preliminary investigation of the need for an IS and proposes a plan to undertake its development. The second activity, *analysis*, seeks to develop requirements for the IS based on the needs of its prospective users and the stakeholders affected by its development. During the *design* activity, the requirements drawn from the analysis are used to create a specification detailing how the IS will function from a technical standpoint. In the *implementation* activity, the system is built and put in production in the organization. Together these four steps represent a *sequence* of generic activities which together contribute to the development of an IS. The actual performance of the ISD process is based the enactment of practices which contribute to those activities.

It is important to note, however, that the SDLC does not impose a particular sequence for its constituting activities beyond the *generic* sequence of planning, analyzing, designing and implementing an IS. Rather, it serves as a canvas against which methods are defined to support the ISD process by formulating normative prescriptions. For example, the ability to perform these activities in an iterative manner is prescribed by methods such as the evolutionary development (Connel & Shafer, 1989), spiral model (Boehm, 1986) or Scrum (Schwaber & Beedle, 2003).

While the four activities of the SDLC provide a useful frame to describe the ISD organizational routine in a general manner, they also are too generic to describe a given ISD process in detail. Indeed, there is no single official description of the SDLC. As a result, the actions that fit within each of its four constituting activities cannot readily be identified. To address this shortcoming, we turn to IS manuals and ISD method books to inventory its composing actions (see Table 4.2). While the selection of the sources presented here is arbitrary, they provide enough overlap to give face validity and support the arguments developed in this work. We also include the description of the Agile SDLC (Ambler, 2006) which presents an alternative representation of the SDLC based on the values advocated by proponents of Agile ISD (Beck et al., 2001) to reflect recent trends in the practice of ISD.

Table 4.2. Detailed Activities of the SDLC

Avison and Fitzgerald (2006)	Dennis et al. (2008)	Valacich et al. (2004)	Ambler (2006)
1. Feasibility study: a. Overview of existing system and gap analysis b. Presentation of potential solutions 2. Systems investigation: a. Detailed investigation of user needs 3. Systems analysis: a. Requirements engineering for the proposed solution 4. Systems design: a. Creation of models and documentation of the expected behavior of the IS and the processes affected by its implementation 5. Implementation: a. Purchase of hardware and software (if required) b. Development, customization of the IS c. Validation of the IS through quality control d. Maintenance of documentation and security procedures e. Operation of new system and de-commissioning of the old IS 6. Review and maintenance: a. Evaluation of the success of the IS and the lessons learned from its implementation b. Implementation of change requests and improvements based on user feedback	1. Planning: a. Identify opportunity b. Analyze feasibility c. Develop work plan d. Staff project e. Control and direct project 2. Analysis: a. Develop analysis strategy b. Determine business requirements c. Create use cases d. Model processes e. Model data 3. Design: a. Design physical system b. Design architecture c. Design interface d. Design programs e. Design databases and files 4. Implementation: a. Construct system b. Install system c. Maintain system d. Post-implementation	1. Systems planning and selection: a. Project identification and selection b. Project initiation and planning 2. Systems analysis: a. Requirements determination b. Requirements structuring c. Alternative generation and selection 3. Systems design: a. Designing the human interface b. Designing databases 4. Systems implementation and operation: a. Coding b. Testing c. Installation d. Documentation e. Training f. Support g. Maintenance	1. Concept (iteration -1): a. Identify potential projects b. Prioritize potential projects c. Develop initial vision d. Consider project feasibility 2. Inception (iteration 0, warm up): a. Obtain funding and support b. Start building the team c. Initial requirements envisioning d. Initial architecture envisioning e. Setup environment 3. Construction of iterations: a. Collaborative development b. Model storming c. Test driven design d. Confirmatory testing e. Evolution of documentation f. Internally deploy software 4. Transition (release, end game): a. Final system testing b. Final acceptance testing c. Finalization of documentation d. Pilot testing of release e. End user training f. Production staff training g. Deploying system into production 5. Production: a. Operate the system b. Support the system

			c. Identify defects and enhancements 6. Retirement: a. Remove the final version of the system b. Data conversion c. Migrate users d. Update enterprise models

We studied the activities described in these sources to conceive a generic depiction of the SDLC by matching activities across sources and grouping them based on their recurrence. This process yielded a total of five generic activities (see Table 4.3) that are in line with practitioners' depictions of the SDLC (e.g., Wikipedia, 2015). We argue that together, these five phases and their constituting actions provide a comprehensive overview of the ISD process and its linked activities carried out by multiple actors. Emphasizing the importance of their performance by organizational actors describing a *process*, we describe them using verbs (Langley, 2007).

Table 4.3. Activities and actions of the SDLC

Main Activity	Actions
Planning	Prioritize projects Select project Develop project plan Staff project
Analyzing	Determine requirements Prioritize requirements Create logical models for requirements
Designing	Determine whether to make or buy the system Convert logical models into physical models Model system architecture Model system behavior
Implementing	Code system Test system Document system Train users and staff Deploy system in production Decommission old system (if any)
Maintaining	Support the system's users Support the system's operation Fix defects Implement enhancements Evaluate system success

We posit that this conceptualization of ISD as a business process made of patterns of interdependent actions provides the foundation to study ISD as an organizational routine based on the ontology proposed by Feldman and Pentland. In the next section, we take this conceptualization further to argue that this ontology allows us to combine elements of the mechanistic and emergent perspectives on ISD.

4.3. Ostensive and Performative Aspects of Organizational Routines

Early conceptualizations of organizational routines (e.g., Cyert & March, 1963; March & Simon, 1958) focused on the *mechanistic* aspect of organizational work and the stability of the patterns of the actions forming an organizational routine. Interestingly, this view matches that of mechanistic perspective on ISD which conceptualizes ISD as a generic pattern of actions repeated over time, as illustrated by the SDLC. In contrast, the ontology developed by Feldman and Pentland provides a means to acknowledge the *emergent* nature of ISD and the role individuals play in its performance – as argued by Agile ISD – incorporating both agency and structuration in the enactment of organizational routines. Their work bridges two different, yet complementary views of social phenomena that had previously been proposed as dichotomous (Latour, 1986). To that end, Feldman and Pentland propose that organizational routines contain two aspects under which they materialize as “effortful accomplishments” (Pentland & Rueter, 1994:488).

4.3.1. The Ostensive Aspect of Organizational Routines

The *ostensive* aspect of an organizational routine is based on a common, shared perception of the routine which remains relatively stable over time. In their work, Feldman and Pentland (2003) use the example of a hiring routine made of a series of actions such as attracting and screening applicants to describe this aspect. In ISD, actors have a shared understanding of what the ISD process entails and against which their performances are enacted. While Pentland and Feldman (2005) argue that the ostensive aspect of an organizational routine may vary based on the perceptions of the actors involved in its performance along with the context within which it exists (e.g., organizational), they also suggest that it may be “fairly coherent” (p. 805) across those contexts. This subtle yet important distinction highlights the possibility for multiple, concurrent ostensive aspects of the same organizational routine to coexist at multiple levels (Feldman & Orlikowski, 2011; Pentland & Feldman, 2005). In ISD, this entails that not all the ostensive aspects of the routine may be congruent. Within a given project, a project manager may consider quality assurance as an activity carried out at the end of the project while developers may consider it part of their everyday activities as they rely on test-driven development (Beck & Andres, 2004) to perform their work. Across projects or organizations, there may also be multiple ostensive aspects that coexist based on the use of different methods which shape the understanding actors form of the ISD process. For example, practitioners accustomed to the waterfall model share a conceptualization of the ISD process anchored in the sequential execution of seven activities (Royce, 1970:329): system requirements, software requirements, analysis, program design, coding, testing, and operations. However, users of the Scrum method do not share this frame of reference.

4.3.2. The Performative Aspect of Organizational Routines

The *performative* aspect of an organizational routine represents the actual enactment of a given action for which actors may exert agency and choose from a "repertoire of possibilities" (Feldman & Pentland, 2003:102) while still being faithful to the ostensive aspect of the routine. While the ostensive aspect of an organizational routine may not exhibit significant *variation* across settings, its performative aspect changes based on the context within which it is enacted, rendering it highly improvisational (Feldman, 2000; Orlikowski, 2000). For example, Feldman (2004) observed the evolution of a hiring routine in a university's residential hall as a result of variations in its performance despite the relative stability of its ostensive aspect. In ISD, the literature offers ample evidence of the need for actors to improvise in their work based on contingencies such as changes in requirements (Lee & Xia, 2010), technology (Lyytinen et al., 2010), or the context within which the artifact is being developed (Fitzgerald et al., 2006; Sarker & Sarker, 2009). More recently, Cecez-Kecmanovic et al. (2014) have studied the development and implementation of an IS in the insurance industry using a performative perspective. Their findings highlight the ability for members of actor networks to create *representations* of the success or failure of an IS in different, and at times conflicting manners. Notwithstanding their insights, we note that this perspective focuses entirely on the concurrent existence of multiple performative views of ISD while forgoing the possibility for those views to be shaped – at least partially – by the existence of multiple ostensive aspects that have the potential to frame varying conceptualizations of success. For example, project managers may consider that successful projects must be delivered on time and on budget while users may conceptualize success based on the delivery of all required functionality, regardless of the budget or schedule.

4.3.3. Artifacts and Organizational Routines

While not part of organizational routines, artifacts play an important role in supporting or challenging actors' performances of a routine as well as their understanding of its structure (Pentland & Feldman, 2005). Artifacts take on a number of shapes, from standard operating procedures, documentation, and training manuals to, more recently, workflow tools (e.g., Pentland et al., 2009; Pentland & Feldman, 2007; Pentland et al., 2011) and other pieces of software that enable and constrain the execution of work by actors. Applied to the ontology of organizational routines as proposed by Feldman and Pentland, this initially suggests that artifacts act as proxies of the ostensive aspect of a routine (Pentland & Feldman, 2005).

However, Pentland and Feldman (2005) and Feldman and Orlikowski (2011) note that artifacts alone cannot represent the multiplicity of ostensive aspects that exist within a given setting. In addition, Becker (2004:652) observes, based on past empirical research on the topic, that transferring routines across settings is a highly complex endeavor and that there may not be "such thing as a universal best practice" regardless of the existence of artifacts that may suggest otherwise. Artifacts however may provide clues regarding a group of actors' desire to enforce a single ostensive aspect of a routine, reflective of the issues of power and symbolism in organizations (Pentland & Feldman, 2008). In the context of ISD, this may allow us to better understand the adoption process of ISD methods, whether it comes from top management or directly from ISD teams, as is often reported in Agile ISD studies (e.g., Atlas, 2009).

Artifacts also have important implications for the study of performances (Pentland & Feldman, 2005). Indeed, artifacts may, through their design and configuration (e.g., D'Adderio, 2003), enable, constrain and ultimately influence the evaluation of work against their features. For example, Pentland and Feldman (2008) report on the failed design and implementation of an IS based on a single group of stakeholders' general understanding of various work processes. As the authors note in conclusion, "Ultimately, it is important to realize that managers design artifacts, not routines. They hope that these artifacts will shape the ostensive aspect of a new routine, and also constrain the performances in some desirable way. But when the participants actually start producing performances, it is not necessarily what the designers had in mind. Some amount of improvisation is inherent in the execution of routines. For better or worse, organizational routines have a life of their own" (p. 249).

4.3.4. Structuration through interaction between ostensive and performative aspects of routines

Building on the ostensive and performative aspects of organizational routines, Feldman and Pentland advocate the study of the relationship between these two aspects, noting the structuration process engaged by their interplay (see Figure 4.2). For instance, the ostensive aspect of an organizational routine provides a codified frame of reference that actors use when referring to a particular process in an organization along with the roles assigned to the actors in charge of executing its constituting activities. For example, Volkoff et al. (2007) studied the implementation of an enterprise system in an organization and found that embedding work processes in technological artifacts gave them a material aspect that mediated the process of change associated with the implementation of IS. This material aspect has an impact on the ostensive and performative aspects of work processes as actors interact with those artifacts to build a shared understanding and enact those processes in practice.

In ISD, business analysts are considered to be primarily responsible for the analysis of the requirements of a piece of software while developers handle its implementation. The generic nature of the SDLC and ISD methods allows actors to use them as an anchor across projects and organizations. The performative aspect of an organizational routine is the enactment of the activities constituting the routine against the *backdrop* of its ostensive aspect. The performance of an organizational routine is "inherently improvisational" (Feldman & Pentland, 2003:102) and rests on the agency of individuals and the contingencies of the context within which they operate. In ISD, the literature provides many examples of actors deviating from the common frame of reference prescribed by a particular approach or method adopted by an organization in order to fit the context where the artifact is developed. For example, Fitzgerald et al. (2006) describe the adoption of agile methods at Intel Shannon in Ireland. Their findings highlight the importance of the context where the ISD process is performed. More specifically, the adoption of methods (in this case, XP and Scrum) is a reflexive process allowing actors to eventually keep some of the practices prescribed by a method while forgoing others, resulting in performances which both deviate and adhere to that method.

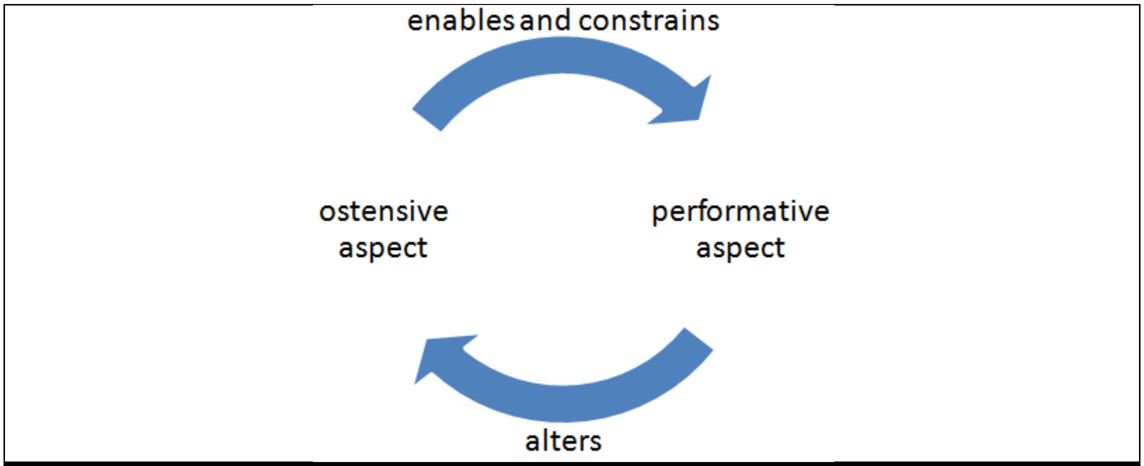


Figure 4.2. The ostensive and performative aspects of organizational routines as elements of structuration

Consistent with arguments from structuration theory (Giddens, 1984), Feldman and Pentland's ontology highlights the constraining and enabling power of the ostensive while variations in performances may, over time, alter the ostensive aspect over time. In ISD, this translates into the ability for actors to perform activities that are in line with the spirit of an ISD method while at the same time using the experience gained from variations in those performances to alter the prescriptions of the method or design entirely new ways of executing work (e.g., Sarker & Sarker, 2009). Indeed, researchers have distinguished between ISD methods as they appear on paper and their instantiation as "methods-in-action", arguing that they are "rarely applied in their entirety, nor as originally intended by their creators, although they may provide a template to guide development practice." (Fitzgerald et al., 2002:13). Similarly, ISD methods such as Scrum or XP are founded on performances enacted in various ISD projects by their authors, illustrating how performances may, over time, lead to the development of an alternative ostensive aspect of the ISD organizational routine. Overall, we argue that a conceptualization of ISD based on the ontology of organizational routines proposed by Feldman and Pentland (2003) allows us to account for three, interrelated, elements. First, the *ostensive aspect* of ISD as a business process which involves a series of generic actions which may be prescribed by a method. Second, the *performative aspect* of ISD and which relies on the practice of ISD by organizational actors. Third, and perhaps most importantly, the interplay between the ostensive and performative aspects as actors, through their performances, both adhere and deviate from the ostensive aspect of ISD.

The arguments presented above point out two important characteristics of the ISD process. First, methods represent artifacts that offer a mechanistic, routine depiction of the ISD process. As organizational actors look to implement those methods, they use them as beacons guiding their performances. Second, the organizational and project-level contexts within which ISD is performed have an important impact on the ability for actors to build and conform to a shared understanding of the ISD routine. For example, there are dozens of studies of ISD projects using popular methods such as XP (e.g., Mangalaraj et al., 2009; Vidgen & Wang, 2009) or Scrum (e.g, Salo & Abrahamsson, 2008; Suscheck & Ford, 2008). Yet, regardless of the method adopted by actors in those studies, the performance of ISD shows a large degree of variation. For example, distributed development and offshore ISD projects provide a context outside of traditional Agile ISD methods' prescriptions. Yet the practices prescribed by these methods are successfully used in these types of projects when they are adapted to the constraints faced by organizational actors (e.g., Sarker & Sarker, 2009).

Overall, we argue that the ostensive and performative aspects of ISD are equally relevant, as illustrated by numerous studies reporting on ISD method customization (e.g., Fitzgerald et al., 2006; Xu & Ramesh, 2007) and other cases detailing the adoption of ISD methods and the oft cited unavoidable improvisations that are required to fit those methods to the context at hand. As we have argued above, a complete focus on the performance of ISD, while insightful, does not provide a comprehensive picture of the ISD process for two reasons. First, it fails to take into account the structuring power of methods as templates used by organizational actors within a specific context. Second, it forgoes the impact of variations in performances as a cause for the evolution of those templates over time.

4.3.5. Methodological implications

The study of organizational routines relies on the collection of longitudinal data that can span the course of weeks (e.g., Pentland et al., 2011) or years (e.g., Salvato, 2009) to provide sufficient data points for the identification of repetitive patterns as well as the evolution of those patterns. The study of the ostensive aspect of a routine requires the researcher to unearth the “generalized understandings” (Pentland & Feldman, 2007:787) of the routine. While artifacts may appear as ideal candidates of this representation, they only provide guidance as to the idea that a particular ostensive aspect is being promoted as ideal or desirable by a group of stakeholders (Pentland & Feldman, 2005). Qualitative research methods, such as interviews and focus groups help researchers elicit this shared understanding and uncover the (potentially multiple) ostensive aspect of the routine. As common patterns emerge from data collected among actors who share a common “point of view” (Feldman & Orlikowski, 2011:1245), the ostensive aspect of the routine takes shape. Studying the performative aspect of a routine often relies on field observation where the actions performed by individuals *in situ* can be properly recorded. Recent works have also advocated the collection of data gathered from workflow tools (Pentland et al., 2009; Pentland & Feldman, 2005, 2007) which can record all the actions performed by actors, whether human or not, in the form of time stamped logs that can later be analyzed, provided that the use of the workflow tools is faithful to the actual pattern enacted in practice.

Data analysis is performed using a wide variety of tools depending on the objectives of the researcher. Organizational routines and their constituting actions undergo processes of variation, selection and retention which explain their evolution over time. In the ontology developed by Feldmand and Pentland, this translates into evolutions in patterns of performances as well as evolutions in the ostensive aspects of routines. This evolutionary process provides a crucial means to reconcile the paradox of stability and continuous change that is characteristic of studies on organizational routines (e.g., Pentland et al., 2012).

Toward that end, several tools have been used by past research on the topic (Becker & Lazaric, 2009). Their number, their diversity and their successful application in contexts where there is a high intensity of innovation (e.g., Salvato, 2009) like ISD are testament to the potential of this perspective to study a variety of organizational phenomena. For instance, Pentland and Feldman (2007) have extended the concept of narrative networks to allow researchers to map and link actions that together constitute a routine’s performance. These networks can be compared over time to uncover stable patterns of actions and deviations from those patterns as a representation of improvisation. Similarly, Pentland’s early work on sequential variety (Pentland, 2003) and grammatical models (Pentland, 1995) provide important tools to define the *domain* within which performances are enacted – the ostensive aspect of a routine – as well as variations across performances. Building on tools used in DNA sequencing in biology, Salvato (2009) proposes the use of Optimal Matching Analysis (OMA) to uncover the structure of routines as well as their evolution by treating them as series of complex event sequences. In order to study the multilevel implications (Salvato & Rerup, 2011) of organizational routines, Pentland et al. (2012) propose the development of a generative model of organizational routines. This model allows researchers to uncover the variation and selective retention of actions over time as a *history* of patterns and is subsequently validated using simulation.

Building on the ontological and methodological features of the perspective proposed by Feldman and Pentland, we propose to illustrate the potential of a conceptualization of ISD as an organizational routine through its application to the context of Agile ISD.

4.4. Ostensive and Performative Aspects of ISD: The Case of Agile ISD

Agile ISD is a topic of growing interest to researchers and practitioners alike (Dingsøyr et al., 2012), as evidenced by the various practitioner conferences on Agile ISD and the presence of special issues in scientific journals (e.g., Abrahamsson et al., 2009; Baskerville, 2006; Dingsøyr et al., 2012). While there are early depictions of “agile” approaches to IS development which date to the late nineteen nineties (e.g., Aoyama, 1997, 1998), the popularity of Agile ISD as a topic of interest to researchers and practitioners follows the publication of the Agile Manifesto at the root of popular methods such as Scrum or XP created by some of its signatories (e.g., Kent Beck, Alistair Cockburn, Jim Highsmith, Ken Schwaber). The Agile Manifesto consists in four core values and twelve principles. Its values are formulated based on a logic of opposition encapsulating some of the most common criticisms made to traditional, plan-based approaches such as the Waterfall model (Royce, 1970). The principles provide general guidelines governing ISD through their reliance on clients and developers as the main actors responsible for the ISD process. Together the values and principles of the manifesto tackle ISD as intrinsically complex and uncertain and propose a malleable approach focused on the rapid and iterative delivery of software increments while accounting for changes in their needs and requirements. Overall the manifesto itself is more akin to a philosophy on the ISD process for which Agile methods provide concrete sets of prescriptive practices that allegedly allow a team of software developers to gain agility through their enactment.

Although authors of agile methods are quick to differentiate their approach to ISD from plan-based approaches (e.g., Beck & Andres, 2004; Schwaber & Beedle, 2003), it has been argued that Agile ISD consists in the same generic activities found in other, plan-based ISD methods and the SDLC in general, although these are performed iteratively across short development cycles (Avison & Fitzgerald, 2006; Erickson et al., 2005). For example XP advocates the use of "test-first programming" (Beck & Andres, 2004:50), a practice wherein code is written against a failed test case that must eventually pass with success. In essence, this practice effectively helps to fulfill both the coding and testing actions of the implementation activities of the SDLC. Overall, authors of agile methods favor lightweight processes to conduct ISD projects in an adaptive rather than predictive manner (Avison & Fitzgerald, 2006), relying on the skills of the people in the ISD team to decide how to manage the development process in the most effective manner given the context.

From an organizational routines perspective, Agile ISD's reliance on the agency of actors to make decisions regarding some of the aspects of the ISD process means that the performative aspect of ISD is of critical importance. However, the shared understanding that actors form as to how the ISD process should be performed – its ostensive aspect – is equally important because it shapes those performances. For example, in the Scrum method, actors should have a shared understanding of the concept of closed iterations where iterations, once started, cannot be altered in response to users' requests to fix some bugs as soon as they are discovered.

4.4.1. Issues with research on Agile ISD

Despite the growing interest in Agile ISD (Dingsøyr et al., 2012) and the advances that have been made on the topic (e.g., Conboy et al., 2011; Lyytinen & Rose, 2006; Strode et al., 2012), research on Agile ISD experiences a number of enduring issues. These issues have been raised in review articles (e.g., Dingsøyr et al., 2008), conceptual works (e.g., Conboy, 2009), as well as empirical research (e.g., Sarker & Sarker, 2009). Building on these works, we group these issues into five overarching themes and present them in Table 4.4. We distinguish between epistemological issues that relate to the definition and acquisition of knowledge (Hirschheim & Klein, 1989) on agile ISD and phenomenological issues that deal with the manifestations of the phenomenon and our ability to study those manifestations using judicious theoretical lenses and methods. We present these issues along with an explanation as to how our conceptualization of ISD as an organizational routine based on the ontology proposed by Feldmand and Pentland helps to address them.

Table 4.4. Issues in Agile ISD research

	Issue	Sub-issue(s)	Main reporting source(s)
Epistemological issues	Lack of strong theoretical groundings for studying Agile ISD	Lack of theoretical glue	Conboy (2009) Conboy and Fitzgerald (2004) Hummel (2014)
		Lack of parsimony	Conboy (2009)
		Limited applicability	Conboy (2009)
	Lack of rigor		Abrahamsson et al. (2009) Dybå and Dingsøyr (2008) McLeod and MacDonell (2011)
Phenomenological issues	Lack of definitional clarity of the concept of agility in ISD		Conboy (2009) Conboy and Fitzgerald (2004) Hummel (2014)
	Lack of cumulative research tradition		Conboy (2009) Dingsøyr et al. (2012)
	Neglect of the long-term impacts of Agile ISD	Focus on the adoption phase	Abrahamsson et al. (2009) Dybå and Dingsøyr (2008)
		Focus on short-term, project-scope benefits	Ashurst et al. (2008)
		Disregard for long-term impacts	Dybå (2002)

Issue 1: Lack of strong theoretical groundings for studying Agile ISD. In his work, Conboy (2009) argues that research on Agile ISD suffers from a “lack of theoretical glue” (pp. 330) because the prescriptions of practices advocated by methods do not explain how, in practice, those practices relate to the achievement of agility. He also notes that many methods “lack parsimony” (pp. 331) because they advocate the enactment of

practices which may be redundant or fail to add value and as a result are rarely used (e.g., XP's system metaphor). Finally, he uses the evidence regarding the frequent tailoring and adaptation of commercial methods to point at their "limited applicability" (pp. 331). We argue that these three issues all relate to an emphasis that is set on commercial methods as overarching artifacts that act as references against which research derives its insights. Given that these methods have emerged from the realm of practice rather than research, they lack the theoretical foundation from which we can build theory on Agile ISD.

Issue 2: Lack of rigor. This issue relates to the lack of methodological (Abrahamsson et al., 2009; Dybå & Dingsøyr, 2008) and theoretical (Erickson et al., 2005; Sarker & Sarker, 2009) rigor associated with many works on Agile ISD. For example, in their systematic review of the literature on Agile ISD Dybå and Dingsøyr (2008) found that only 36 out of 270 empirical works met their inclusion criteria of quality and rigor. In a similar vein, many works report on experiences, lessons learnt and challenges (e.g., Nerur et al., 2005) where the evidence is often self-reported (e.g, Block, 2011) and analyzed in an atheoretical (e.g., Berczuk & Lv, 2010) or incomplete manner (e.g., Ashurst et al., 2008; Whitworth & Biddle, 2007), thereby limiting the insights gained from the evidence presented. In many of those accounts, the core artifact studied by researchers is the use of an Agile method within a particular context.

Issue 3: Lack of definitional clarity of the concept of agility in ISD. In a systematic review of the literature on Agile ISD Hummel (2014) notes that 80% of the works surveyed failed to define the concept of agility in ISD while many others used the Agile Manifesto's values and principles as its proxy, observing that "the definition of agility remains one of the most salient problems of Agile ISD" (p. 4718). While commercial Agile methods claim to base the values, principles, and practices they promote on the notion of agility advocated by the manifesto, it is unclear what is meant exactly by the term "agility" in either of these artifacts. Indeed, without a proper definition as to what the term "agile" means, it acts as a *placeholder* more than a concept that can be effectively used by practitioners, let alone researchers.

Issue 4: Lack of cumulative research tradition. In many works, the use of an Agile method is studied from an adaptation or tailoring perspective where actors work to bend the values, principles and practices of a formal method to fit the context where it must be employed, for example through the selective enactment of prescribed practices (e.g., Aydin et al., 2004; Cao et al., 2009; Conboy & Fitzgerald, 2010; Srinivasan et al., 2009). While this approach has merit in its proximity to the realm of practice, it fails to build insightful knowledge on the relevance of practices *in a cumulative fashion* across settings by anchoring them with fundamental concepts situated outside the definition of methods, such as agility or flexibility (Conboy, 2009). As a result, these works invariably depict deviations and variations from a well-known, prescribed pattern whereas an evolutionary approach across multiple cases could help derive insights that better serve the goal of building a theory on Agile ISD (Conboy, 2009). The other consequence of this issue is the seemingly revolutionary aspect of Agile ISD which sometimes appears as a complete departure from previous ISD approaches. However, some have rebutted this argument (e.g., Turner & Boehm, 2003) by stressing that "commentators fail to realize that most agile practices have origins in much older methods" (Conboy, 2009:331), such as the spiral model (Boehm, 1988) and other evolutionary development approaches (Avison & Fitzgerald, 2006). Overall, the study of Agile ISD as a method being used as a novel artifact prevents us from inscribing it within the traditional ISD paradigm, thereby limiting our potential to build cumulative knowledge.

Issue 5: Neglect of the long-term impacts of Agile ISD. The logic of opposition and the emphasis used to state the four core values of the Agile Manifesto (e.g., "**responding to change** over following a plan") portray Agile ISD as a significant departure from traditional, plan-based-approaches. In line with this view, the authors of Scrum (Schwaber & Beedle, 2003) note that "basically, practicing Scrum changes the culture of an organization because Scrum comes with new values, beliefs, languages, rules, roles, and practices […]. However, changing cultures is one of the most difficult changes one can ever make" (pp. 118-119). Works in organization studies have long acknowledged the need to alter the values of individuals to trigger changes in organizational behaviors in order to successfully alter the culture of an organization (Cameron & Quinn, 2011). While Agile ISD may be considered as a continuation of the tradition of evolutionary methods, the evidence presented in empirical research suggests that the *adoption* of Agile ISD does indeed represent a process of change. For example, Kim and Ryoo (2012) report the progress of a large organization's transition from a plan-based approach to Scrum and find that after three years, they are "still very far from perfect", arguing that "one of the hardest parts of introducing agile methods into an organization is the cultural change it causes" (p. 481). While research on the long-term impacts of Agile ISD adoption is still scarce (e.g., Passos et al., 2012) and typically anecdotal (e.g., Chung & Drummond, 2009; O'Connor, 2011) despite some notable exceptions (e.g., Vidgen & Wang, 2009), calls have been made to focus on the human and social aspects of this phenomenon as it becomes more mature (Dybå, 2002; Dybå & Dingsøyr, 2008; Müller et al., 2010; Passos et al., 2012).

Overall we argue that the conceptualization of ISD as an organizational routine has the potential to help address some of the most enduring issues on research in agile ISD, as summarized in Table 4.5.

Table 4.5. Addressing Issues in Agile ISD Research through a Conceptualization of ISD as an Organizational Routine

Issue	How organizational routines help to address this issue
Lack of strong theoretical groundings for studying Agile ISD	Organizational routines are a strong theoretical foundation for the study of organizational phenomena that contribute to an organization's survival. Conceptualizing ISD as an organizational routine allows us to inscribe Agile ISD within this stream of research.
Lack of rigor	The existence of a large body of works studying organizational phenomena with this perspective promotes rigor with respect to the assumptions and features of this perspective. Empirical as well as conceptual works provide a number of methodological clues to study the ostensive and performative aspects of organizational routines, their relationship as well as their evolution.
Lack of definitional clarity of the concept of agility in ISD	Organizational routines promote the study of both ostensive and performative aspects guiding the actions of actors. As a result, they offer insight regarding the ISD process in principle (the practices promoted by a method) as well as in practice (the method in action). These two aspects provide a comprehensive representation of the reality of ISD, regardless of the method being used, thereby helping to reach for the essence of the concept of agility in ISD..
Lack of cumulative research tradition	Organizational routines do not focus on the study of the features of methods. Rather, they emphasize the relationship between ostensive and performative aspects that can be compared across settings, regardless of the method used, if any, thereby helping to build a cumulative knowledge base on Agile ISD.
Neglect of the long-term impacts of Agile ISD	Organizational routines promote the study of (1) stability through the existence of structures imposed by their ostensive aspect and (2) change through variations in their performance, opening the door to studying the long term impacts of Agile ISD.

4.5. Avenues for Research on ISD using Organizational Routines in the Context of Agile ISD

It has been suggested that the features of organizational routines, whether taken individually or in relation with one another, provide different perspectives that have the potential to yield insight into the processes studied by researchers as well as their evolution (Pentland & Feldman, 2005). In this section, we follow these suggestions and adapt them to the context of Agile ISD to present six main avenues for research summarized in Table 4.6.

Table 4.6. Avenues for Research in the Context of Agile ISD

Avenue	Expected Insight
1 – Focus on the performance of Agile ISD	• Understand how the context within which Agile ISD projects are undertaken influence the patterns of performances observed in those projects • Identify patterns of performances tending to specific facets of the concept of agility in ISD • Understand performances in Agile ISD projects from a multilevel perspective: (1) as micro-level foundations that drive performances by individuals who adhere to the values and principles of the agile manifesto; (2) as individual-level performances that contribute to the achievement of organizational performance
2 – Focus on the ostensive aspect of Agile ISD	• Understand the emergence, evolution and coexistence of multiple ostensive aspects in Agile ISD projects. • Identify overarching dimensions of Agile ISD to compare and contrast methods despite their varying foci
3 – Focus on the role of artifacts in Agile ISD	• Provide important points of reference guiding the collection and analysis of data pertaining to the ostensive aspect of Agile ISD within the context of a given project
4 – Focus on the interaction between the ostensive and performative aspects of agile ISD	• Understand the long-term implications associated with the adoption of Agile ISD methods based on the evolution of performances as a result of changes in the ostensive aspect of Agile ISD • Uncover important contextual variables that may drive the adoption of Agile ISD methods as bottom-up or top-down processes
5 – Focus on the interaction between performances and artifacts in Agile ISD	• Understand how artifacts may be designed to enforce a specific ostensive aspect, thereby shaping and constraining performance of Agile ISD (e.g., workflow tools)
6 – Focus on the interaction between the ostensive aspect and artifacts in Agile ISD	• Understand how artifacts may, or may not, promote a single ostensive aspect of Agile ISD • Understand how existing ostensive aspects may shape the perception of artifacts by organizational actors involved in Agile ISD projects

4.5.1. Avenue 1: Focus on the Performance of Agile ISD

Research on organizational routines has acknowledged the importance of the context where performances are enacted to identify important variables that influence those performances (Becker, 2004). Indeed, while an organizational routine may seem to indicate that performances *should be* replicable across time and settings, it is seldom the case (Becker, 2004). In the context of Agile ISD, these arguments are especially relevant given the different foci of Agile ISD methods (e.g., Scrum is closer to project management while XP is closer to day-to-day management of developers' tasks), the frequent tailoring and customization of those methods (e.g., Conboy & Fitzgerald, 2004; Xu & Ramesh,

2007) and the importance of self-organizing teams and improvisation (e.g., Beck & Andres, 2004). For instance, several studies report the need to adapt Agile ISD methods in contexts that depart from the original prescriptions of those methods.

The performative aspect of the ISD organizational routine promotes the inclusion of the context as an integral part of the theoretical and empirical analysis to understand how and why performances take certain shapes in certain cases and not others. For instance, comparing performances within a given project, across projects and organizations enables the identification of patterns and key contextual variables that influence performances. For example, coding a new feature and fixing a bug may exhibit very different patterns of performances because the *context* defined by the type of task at hand is different. Similarly, the composition and size of an ISD team may have an impact on the shape of the narrative networks describing performances over time. For instance, if systems analysis is performed by developers in tandem with end users as advocated by Agile ISD methods, the shapes of the narrative networks that depict those performances may differ significantly from those observed in a context where business analysts act as interfaces between developers and end users.

In addition, the identification of those patterns carries the potential to build a better understanding regarding the notion of agility in ISD beyond the adoption and usage of a given method. Indeed, in line with arguments promoting agility in ISD as a multifaceted concept (Sarker et al., 2009), the study of patterns of performances across settings can help identify *patterns* of agility in ISD in the form of *configurations* wherein specific facets are enabled in different contexts, whether due to the use of different Agile ISD methods or specific contextual variables that have a direct impact on those performances (e.g., distributed development). For example, within a given project, patterns associated with lower performances may be identified and corrected to promote more optimal patterns that allow actors to react quickly to changes in requirements.

From a multilevel perspective, studying Agile ISD performances from an organizational routines perspective promotes a better understanding of their impacts at the organizational level. Organizational routines are collective entities that contribute to the realization of an organization's objectives, whether successfully or not. Conceptually, they are a multilevel phenomenon although past research has tended to overlook the implications of this important feature (Salvato & Rerup, 2011). While organization studies have identified a number of important constructs that contribute to an organization's performance, little is known regarding the emergence process (Klein & Kozlowski, 2000) that provides the crucial links between those levels (Salvato & Rerup, 2011).

As a focal construct, organizational routines may be studied as the outcome of individual level performances or the antecedent of organizational level outcomes. For example, the Agile Manifesto and popular methods such as XP or Scrum advocate values and principles which allegedly drive the enactment of certain ISD practices. In order to achieve this objective, actors must experience a process of change that significantly alters their perspective on ISD (Beck & Andres, 2004; Schwaber & Beedle, 2003). Research in agile ISD adoption may therefore look into the changes in the values and principles followed by individuals that lead to variations in the performance of ISD, effectively gaining insight on the micro-foundations of the ISD organizational routine. In this case, we may find that the emergence of those performances is based on a process of *composition* (Klein & Kozlowski, 2000:16) wherein the adherence to common values leads to the emergence of the collective phenomenon. Conversely, one may find that a process of *compilation* (Klein & Kozlowski, 2000:16) leads to the emergence of an organizational routine through a disparity in the adherence to the values and principles of the manifesto.

As a contributing factor of organizational performance, the study of performances of Agile ISD can shed light on impacts that go beyond immediate, project-level outcomes. Like other organizational routines, Agile ISD may help build dynamic capabilities (Eisenhardt & Martin, 2000; Teece et al., 1997), interpretive schema, identity, and strategies at the organizational level (Salvato & Rerup, 2011). Indeed, as a form of *artful planning* (Baskerville, 2006) built around the notion of change and disruption, agile ISD may directly contribute to an organization's ability to quickly sense and respond to changes in their environment (e.g., technology, user requirements) through a swift reconfiguration of their resources, although there is currently little empirical evidence to support this argument. In one paper, Mathiassen and Vainio (2007) studied the contribution of ISD activities in two small software firms to sense and respond to changes emerging from the need to satisfy multiple customers using limited resources. Overall, certain *configurations* of performances of Agile ISD may enable a firm's employees to collectively sense and respond to change, thereby helping to build a sustainable competitive advantage.

4.5.2. Avenue 2: Focus on the Ostensive Aspect of Agile ISD

In their work, Pentland and Feldman (2005) observe that researchers may be tempted to assimilate the ostensive aspect of an organizational routine as a single, uniform understanding as to what actions constitute the routine in the eyes of organizational actors, especially in cases where researchers observe the routine from the perspective of uninformed outsiders. This is especially true in cases where artifacts are prominent and may promote a single vision of the routine. Pentland and Feldman (2005) note however, that the coherence of an organizational routine's ostensive aspect may only be observable on the surface. At a closer look, researchers may discover that there are in fact multiple, concurrent ostensive aspects that exist based on the roles that actors play in the enactment of the routine. Similarly, organizational actors who are external to the performance of the routine may also have built their shared understanding of the routine.

While the existence of multiple ostensive aspects renders the study of organizational routines more complex for researchers, it also opens the door to reconciling the idea that there may be a general pattern of actions that exists and yet varies across categories of actors. In the context of ISD and more specifically Agile ISD, this is especially relevant because Agile ISD promotes multidisciplinary teams composed of actors that represent various types of stakeholders in organizations (e.g., developers, project managers, users). Inevitably, the understanding that these stakeholders have of the ISD process is skewed based on their specialties and skills. For example, the ostensive aspect of ISD for a developer will be more focused on the technical aspects of the development process than an end user. Understanding how these multiple ostensive aspects come into existence, how they evolve and more importantly, how they coexist within the context of a project, as well as across projects and organizations allows researchers to move away from Agile ISD methods as monolithic artifacts that are understood and ultimately performed uniformly by actors.

An interesting application of this avenue would be to reconstruct and contrast, within a given project, the ostensive aspects that coexist and understand how they complement or conflict with one another to understand how actors form expectations of one another and whether these expectations are met in the actual performance of Agile ISD along with the impacts they may have on the project at hand. Across projects, the existence of multiple ostensive aspects may be informed by the adoption and use of different Agile ISD methods. Reconstructing these understandings of the ISD process based on the use different methods may give us the ability to compare those methods using a set of generic dimensions despite the fact that the foci of those methods vary significantly.

4.5.3. Avenue 3: Focus on the Role of Artifacts in Agile ISD

While artifacts are not part of the ostensive and performative aspects of organizational routines, they play an important role to support or consolidate those aspects. In the context of Agile ISD, artifacts may provide a convenient means to gather large amounts of data on organizational routines at a relatively low cost and more importantly, without requiring the physical presence of the researcher on site. As Pentland and Feldman (2005) note, artifacts are relatively stable and provide a tangible means to identify an organizational routine. For researchers, artifacts provide an anchor against which data on the ostensive and performative aspect of an organizational routine may be analyzed. In some cases, artifacts are built to reflect an ostensive aspect of the routine, as can be the case with documentation, rules and standard operating procedures or, in the case of Agile ISD, Agile ISD methods. In other cases where artifacts are used by actors interactively, they help to keep a memory of the performances of actors as they use them to consign their actions.

Notwithstanding the attractive, tangible (whether physically or virtually in the case of software) nature of artifacts, Pentland and Feldman (2005) caution researchers to ensure that they indeed reflect the ostensive or the performative aspect of an organizational routine. In the case of Agile ISD, we argue that artifacts alone provide little insight into the organizational routine itself. Rather, as we develop below, it is the interactions between artifacts and the aspects of organizational routines that provide strong potential for insight on this phenomenon.

4.5.4. Avenue 4: Focus on the Interaction between the Ostensive and Performative Aspects of Agile ISD

In the ontology developed by Feldman and Pentland, the ostensive and performative aspects are mutually constitutive. Accordingly, the ostensive aspect provides a frame of reference against which performances occur, accounting for both the stability and evolution of organizational routines. From a structuration perspective, the ostensive aspect provides a structure against which actors' agencies are enacted, allowing this structure to evolve over time. In Agile ISD, one of the most prominent topics of research in recent years has been the adoption of Agile methods (Dingsøyr et al., 2012) and the changes organizations go through during this process. To date however, a large part of this literature has focused on the practice of ISD and has exhibited a lack of methodological and theoretical rigor that prevents the building of cumulative knowledge on the topic (Conboy, 2009; Dybå & Dingsøyr, 2008; Hummel, 2014). We argue that an organizational routines' perspective on Agile ISD provides a rigorous theoretical frame through which the changes associated with the adoption of Agile methods can be studied as an evolutionary process of organizational change that is the result of the performances enacted by actors within the frame defined by the ostensive aspect of the ISD organizational routine.

An interesting avenue for research would be to study how the aspects of the ISD organizational routine evolve over time along with the impacts associated with those evolutions. For instance, the literature on Agile ISD often depicts Agile ISD adoption as a bottom-up process driven by ISD teams (e.g., Atlas, 2009). These teams alter the contents of their performances in order to modify the ostensive aspect of ISD within an organization. In many cases however, making these changes "stick" and ultimately altering the ostensive aspect of ISD organizational routine proves to be a long and challenging process (e.g., Kim & Ryoo, 2012). Conversely, some studies report success in the adoption of Agile ISD methods when these changes are promoted as a top down process and performances are altered after the current ostensive aspect of the ISD routine is destabilized by the official adoption of an Agile ISD method (e.g., Chung & Drummond, 2009). Studying the evolution of those processes of change may help us account for important contextual variables that promote bottom up or top down processes of adoption and improve the success of those initiatives.

Studying the adoption of Agile ISD methods, we may also find that some actions of the ISD organizational routine are simply left unchanged and study the underlying causes of this stability. This may help uncover some interesting insights regarding the lack of duality between traditional and agile approaches as argued by some authors (e.g., Turner & Boehm, 2003) as well as the presence of certain elements such as power or political pressures that prevent changes from being implemented as originally planned.

Finally, we may discover other organizational routines that are deeply affected by Agile ISD adoption even though they tend to be regarded as outside the scope of the phenomenon. For example, it has been noted that changes must be brought to processes such as budgeting, compensating or hiring when adhering to agile values and principles (e.g., Beck & Andres, 2004; Cohn & Ford, 2003; Schwaber & Beedle, 2003) as they have an impact on the ISD process itself. As a phenomenon of interest, the definition of an given organizational routine is based on the interests of the researcher and may therefore extend beyond the activities that are traditionally prescribed in Agile ISD methods and which may tend to focus too much on the project management or technical activities of ISD.

4.5.5. Avenue 5: Focus on the Interaction between Performances and Artifacts in Agile ISD

In line with the central role played by technology in shaping the performance of ISD, artifacts play an important role in the context of Agile ISD. While these artifacts may often be technological and take the shape of workflow and project management tools (e.g., Atlassian, 2015), Agile ISD has long promoted the use of physical artifacts such as whiteboards and sticky notes to manage a project. In this section, we focus on the role of workflow tools used to manage Agile projects (VersionOne.com, 2013) to study the interaction between artifacts and the performances of actors.

Workflow tools define a series of actions that actors must follow when they perform their tasks. These tools can further be configured to force certain transitions (e.g., a programming task cannot be sent to quality assurance until it has been peer reviewed), require particular pieces of information (e.g., closing a work item requires a comment, committing source code requires a link to a work item) as well as grant or restrict access to performing certain actions based on the roles of actors in the project (e.g., only a project manager can close a release). These configurations effectively constrain and shape the array of performances available to actors in their enactment of the routine. Another interesting implication of the adoption of those tools is the level of integration that exists between them. For example, XP promotes the use of *continuous integration* (Beck & Andres, 2004), a practice wherein the source code must be ready to deploy and test in a matter of minutes at all times. To facilitate this process, workflow tools can integrate with other tools, as illustrated in Figure 4.3. The level of integration configured in those tools deeply influences the performance of work by project members as they may have to search for information and perform some operations manually (e.g., building the artifact on a separate build system) in some cases while they may be automated and effectively restrict the amount of improvisation available to actors.

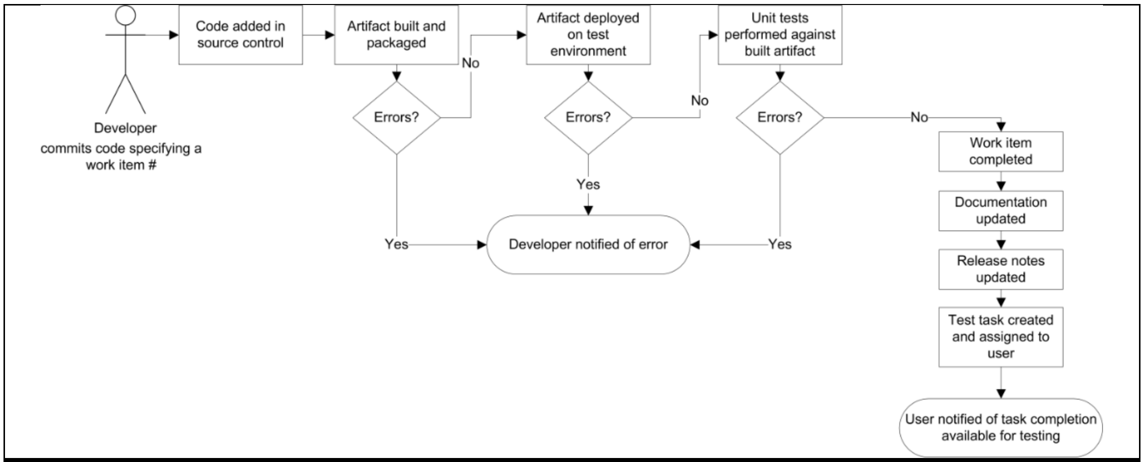

Developer
commits code specifying a work item #
Code added in source control
Artifact built and packaged
Errors?
No
Yes
Artifact deployed on test environment
Errors?
No
Yes
Unit tests performed against built artifact
Errors?
No
Yes
Developer notified of error
Work item completed
Documentation updated
Release notes updated
Test task created and assigned to user
User notified of task completion available for testing

Figure 4.3. Automating workflow based on the principles of continuous integration

The examples described above show the power of artifacts in shaping the performances of actors in the enactment of organizational routines. In Agile ISD, performances are expected to be improvisational while at the same time, organizations seek to structure the ISD process in order to achieve economies of scale and accumulate knowledge over time. How and why these artifacts are configured and used differently in different contexts may help us understand how and why performances are enacted in certain manners and highlight the role of workarounds when artifacts hinder the performance of the routine (Goh et al., 2011). From a methodological perspective, artifacts may provide a way to triangulate performance data. In addition, workflow tools provide a means for actors to translate their actions in real life into series of virtual steps. Building on the word of caution provided by Pentland and Feldman (2005) on the use of workflow data to analyze performances, an interesting avenue for research would be to look into the discrepancies and similarities that may exist between the actual performance of an action and its consignment in a workflow tool. For example, a developer may design a feature, code it and test it with a peer in a single sequence while a workflow tool may treat those as three separate actions, each with different properties and timestamps. A developer may simply complete these three actions one after the other in the tool and enter one large amount of time spent (e.g., four hours) while she may have actually spent one hour on the first action, two on the second and one on the third.

4.5.6. Avenue 6: Focus on the Interaction between the Ostensive Aspect and Artifacts in Agile ISD

While artifacts may constrain or enable the performance of an organizational routine, they also shape its ostensive aspect. Pentland and Feldman (2005:807) note that in many cases, researchers ignore those interactions because artifacts are supposed to be faithful to the ostensive aspect of the routine. However, the coexistence of multiple ostensive aspects among the actors involved in the enactment of an organizational routine means that artifacts may not always be aligned with those ostensive aspects. Indeed, project documentation or workflow tool configurations may reflect the ostensive aspect of project management staff while developers and end users may have a different understanding of the ISD process. In line with some arguments developed in institutional theory (Scott, 2008), artifacts are used to *carry* a preferred pattern of action. Those who control the design and configuration of those artifacts effectively hold power over other actors and can enforce various types of control mechanisms to see that their ostensive aspect of the routine prevails. For example, workflow tool configuration may enable behavior control if it forces actors to go through certain transitions in their daily tasks. While performances would certainly be affected by this configuration, one may wonder whether over time, the shared understanding of the routine itself could be affected. An interesting related question on the topic is the study of the role of artifacts on the emergence of ostensive aspects that deviate from the one promoted by those artifacts.

While artifacts are typically studied as objective, crystallized entities, their interaction with the ostensive aspect of an organizational routine is based on their *perception* by actors. Indeed, research in IS has long acknowledged the role of perception in the adoption and usage of technology. In the case of workflow tools, an interesting avenue for research would be to study how varying perceptions of artifacts and their affordances shape the existence and degree of congruence of multiple ostensive aspects of the ISD organizational routine. For example, project managers who have complete access to the entire set of operations enabled in a workflow tool may consider that the ISD organizational routine comprises a wide range of actions. Conversely, end users may only have access to a subset of those operations and build an understanding that the ISD organizational routine focuses on developing functionality, testing it and finding defects. Studying the *fit* between artifact configurations and the shared understanding of the actors involved in the enactment of the organizational routine may reveal discrepancies that account for alternate representations of those artifacts. Alternatively, one may consider this interaction in reverse and study the role of ostensive aspects in shaping the relationship between actors and artifacts. In the case of workflow tools, an interesting avenue would be to study how patterns of usage of those tools vary based on the ostensive aspect of the ISD organizational routine.

4.6. Conclusion

In this work, we built on two main perspectives on ISD and argued that they each reflect important characteristics of this phenomenon. On the one hand, actors can be expected to form expectations as to how an ISD project should unfold in practice based on a certain history or the prescriptions of the method. On the other hand, project contingencies and the agency of organizational actors translate into variations in practice from those expectations.

We proposed the ontology of organizational routines (Feldman & Pentland, 2003) to account for these two aspects as well as their relationship. More specifically, we argued that ISD is a business process that fits within the paradigm of organizational routines. As an organizational routine, ISD is composed of an ostensive aspect, reflecting the “generalized idea” (Feldman & Pentland, 2003:101) actors have of its unfolding and a performative aspect encompassing the actual performances of actors. While the ostensive aspect of ISD may be relatively stable and even coherent among actors in spite of its multiplicity, performances are “inherently improvisational” (Feldman & Pentland, 2003:102). The relationship between these two aspects accounts for the ability for the activities of ISD to evolve over time through variations of its performance. We further acknowledged the role of artifacts as carriers of ostensive aspects as well as enabling and constraining forces of performances.

We then applied this conceptualization to the Agile ISD phenomenon, one of the most current and relevant area of interest for researcher (Dingsøyr et al., 2012) and practitioners (VersionOne.com, 2013) alike. We built on extant research and highlighted five overarching issues on the topic. To demonstrate the usefulness of our conceptualization of ISD and help address those issues, we developed six avenues for research on Agile ISD.

The first avenue advocates the study of performances to better understand the influence of the context Agile ISD is used, the notion of agility in ISD as well as the multilevel implications of the foundations and organizational impacts of Agile ISD. The second avenue focuses on the ostensive aspect of Agile ISD, building on the multiplicity of ostensive aspects within a group of organizational actors to understand their congruence – or lack therefore – in a given project or organization. The third avenue showcases the role of artifacts in Agile ISD. Rather than arguing for their study as independent elements we argue that it is their relationship with the ostensive and performative aspects of organizational routines that has the potential to yield the most significant insight. The fourth avenue focuses on the interaction between the ostensive and performative aspects of ISD to uncover patterns of evolution and long term impacts of Agile ISD adoption. The fifth avenue deals with the role of artifacts as enabling and constraining forces of performances in Agile ISD. The final avenue proposes to study the relationship between artifacts and the ostensive aspect of the ISD organizational routine, more specifically the fit – or lack of fit – between some of ostensive aspects that are coexisting within a given ISD project that are crystallized in the configuration of those artifacts.

This work makes its main contribution to the literature on ISD. We propose a conceptualization of ISD that focuses on the study of the relationships that exists between the idea actors have as to how ISD should unfold in principle, its performance, as well as the artifacts actors use when referring to these two aspects. While those relationships are relevant in the context of ISD method adoption they also remain relevant beyond this initial adoption stage, where research is currently lacking (Abrahamsson et al., 2009; Dingsøyr et al., 2012). Under this conceptualization of ISD, the adoption of an ISD method is but one of the events that punctuate the evolution of the ISD process over time. Offering a conceptualization that is cognizant yet remains independent of methods will allow research to explore previously uncharted territory, something which is especially relevant now that some organizations have a long *history* of IS development.

Another contribution made to the literature on ISD deals with the ability to build a better understanding of the perspectives of multiple stakeholders involved in an ISD project. Indeed, the ontology of organizational routines used in this work acknowledges the possibility for multiple ostensive aspects to coexist. In ISD, this may translate into new insight with regards to the expectations different types of stakeholders (e.g., project managers, developers, users) may form of the ISD process. In practice, those expectations may result in variations in performances that can help explain how stakeholders make sense of the ISD process and further extract the source of issues commonly faced in ISD projects, such as conflicts (Robey et al., 1993) or power struggles (Myers & Young, 1997).

Finally, this work contributes to the literature on Agile ISD. The avenues developed provide clear guidelines for future research on Agile ISD that can help uncover new insight on this topic. In addition, the ontology of organizational routines used in this work can help address some of the most enduring issues faced by research on the topic (Conboy, 2009; Dingsøyr et al., 2012). First, it provides a strong theoretical foundation along with a series of methodological implications that can help build theoretically and methodologically rigorous research on Agile ISD. Second, it enables comparisons by focusing on the ostensive and performative aspects of ISD regardless of the agile ISD method being used. Third, it may help uncover patterns of performances across a variety of settings (including methods) that will contribute to a better understanding of the concept of agility in ISD.

Together, the arguments developed in this work build on our cumulative knowledge of the ISD phenomenon as well as conceptual and methodological advances made in research on organizational routines to encourage researchers to embrace what may be seen as the paradoxical nature of ISD: a process which is fraught with complexity and uncertainty and yet, is part of the daily lives of organizational actors involved in its undertaking.

References

ABRAHAMSSON P, CONBOY K and XIAOFENG W (2009) 'Lots done, more to do': The current state of agile systems development research. *European Journal of Information Systems, 18*, 281-284.

AMBLER S (2006) The agile system development life cycle (sdlc). http://www.ambysoft.com/essays/agileLifecycle.html, accessed December 15, 2014.

ANG S and BEATH CM (1993) Hierarchical elements in software contracts. *Journal of Organizational Computing and Electronic Commerce, 3*(3), 329-361.

AOYAMA M (1997) Agile software process model. In *Computer Software and Applications Conference*, pp. 454-459, Washington, DC.

AOYAMA M (1998) Web-based agile software development. *IEEE Software, 15*(6), 56-65.

ASHURST C, DOHERTY NF and PEPPARD J (2008) Improving the impact of IT development projects: The benefits realization capability model. *European Journal of Information Systems, 17*(4), 352-370.

ATLAS A (2009) *Accidental adoption: The story of scrum at amazon.Com*. Paper presented at the Agile Conference, Chicago, IL.

ATLASSIAN (2015) Jira - issue & project tracking software. https://www.atlassian.com/software/jira, accessed March 2, 2015.

AVISON D and FITZGERALD G (2006) *Information systems development: Methodologies, techniques & tools*. McGraw-Hill Education, New York, NY.

AYDIN MN, HARMSEN F, SLOOTEN KV and STAGWEE R (2004) An agile information systems development method in use. *Turk J Elec Engin, 12*(2), 127-138.

BASKERVILLE RL (2006) Artful planning. *European Journal of Information Systems, 15*(2), 113-115.

BECK K and ANDRES C (2004) *Extreme programming explained: Embrace change*. Addison-Wesley Professional, Upper Saddle River, NJ.

BECK K, et al. (2001) Manifesto for agile software development. http://agilemanifesto.org/, accessed March 1, 2014.

BECKER MC (2004) Organizational routines: A review of the literature. *Industrial and Corporate Change, 13*(4), 643-678.

BECKER MC and LAZARIC N (2009) *Organizational routines: Advancing empirical research*. Edward Elgar Publishing, Northampton, MA.

BERCZUK S and LV Y (2010) We're all in this together. *IEEE Software, 27*(6), 12-15.

BLOCK M (2011) *Evolving to agile: A story of agile adoption at a small saas company*. Paper presented at the Agile Conference, Salt Lake City, UT.

BOEHM B (1986) A spiral model of software development and enhancement. *SIGSOFT Softw. Eng. Notes, 11*(4), 14-24.

BOEHM BW (1988) A spiral model of software development and enhancement. *Computer, 21*(5), 61-72.

BRINKKEMPER S (1996) Method engineering: Engineering of information systems development methods and tools. *Information and software technology, 38*(4), 275-280.

BROOKS FP, JR. (1987) No silver bullet essence and accidents of software engineering. *Computer, 20*(4), 10-19.

CAMERON KS and QUINN RE (2011) *Diagnosing and changing organizational culture: Based on the competing values framework*. John Wiley & Sons, San Francisco, CA.

CAO L, MOHAN K, XU P and RAMESH B (2009) A framework for adapting agile development methodologies. *European Journal of Information Systems, 18*(4), 332-343.

CECEZ-KECMANOVIC D, KAUTZ K and ABRAHALL R (2014) Reframing success and failure of information systems: A performative perspective. [Article]. *MIS Quarterly, 38*(2), 561-588.

CHUNG M-W and DRUMMOND B (2009) *Agile at yahoo! From the trenches*. Paper presented at the Agile Conference, Chicago, IL.

COHN M and FORD D (2003) Introducing an agile process to an organization. *Computer, 36*(6), 74-78.

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

CONBOY K, COYLE S, XIAOFENG W and PIKKARAINEN M (2011) People over process: Key challenges in agile development. *IEEE Software, 28*(4), 48-57.

CONBOY K and FITZGERALD B (2004) *Toward a conceptual framework of agile methods: A study of agility in different disciplines*. Paper presented at the ACM Workshop on Interdisciplinary Software Engineering Research, Newport Beach, CA.

CONBOY K and FITZGERALD B (2010) Method and developer characteristics for effective agile method tailoring: A study of xp expert opinion. *ACM Transactions on Software Engineering and Methodology, 20*(1), 2-30.

CONNEL J and SHAFER L (1989) *Structured rapid prototyping: An evolutionary approach to software development*. Yourdon Press, Prentice Hall Building, Upper Saddle River, NJ.

CYERT RM and MARCH J, G (1963) *A behavioral theory of the firm*. Prentice-Hall, Englewood Cliffs, NJ.

D'ADDERIO L (2003) Configuring software, reconfiguring memories: The influence of integrated systems on the reproduction of knowledge and routines. *Industrial and Corporate Change, 12*(2), 321-350.

DAVENPORT TH (2013) *Process innovation: Reengineering work through information technology*. Harvard Business Press.

DAVIS GB (1982) Strategies for information requirements determination. *IBM Systems Journal, 21*(1), 4-30.

DENNIS A, WIXOM BH and ROTH RM (2008) *Systems analysis and design*. John Wiley & Sons, Hoboken, NJ.

DINGSØYR T, DYBÅ T and ABRAHAMSSON P (2008) *A preliminary roadmap for empirical research on agile software development*. Paper presented at the Agile Conference, Toronto, Canada.

DINGSØYR T, NERUR S, BALIJEPALLY V and MOE NB (2012) A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software, 85*(6), 1213-1221.

DYBÅ T (2002) Enabling software process improvement: An investigation of the importance of organizational issues. *Empirical Software Engineering, 7*(4), 387-390.

DYBÅ T and DINGSØYR T (2008) Empirical studies of agile software development: A systematic review. *Information and software technology, 50*(9/10), 833.

EISENHARDT KM and MARTIN JA (2000) Dynamic capabilities: What are they? *Strategic Management Journal, 21*(10-11), 1105-1121.

ERICKSON J, LYYTINEN K and SIAU K (2005) Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management, 16*(4), 88-100.

FARAJ S and SPROULL L (2000) Coordinating expertise in software development teams. *Management Science, 46*(12), 1554-1568.

FELDMAN MS (2000) Organizational routines as a source of continuous change. *Organization Science, 11*(6), 611-629.

FELDMAN MS (2004) Resources in emerging structures and processes of change. *Organization Science, 15*(3), 295-309.

FELDMAN MS and ORLIKOWSKI WJ (2011) Theorizing practice and practicing theory. *Organization Science, 22*(5), 1240-1253.

FELDMAN MS and PENTLAND BT (2003) Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly, 48*(1), 94-118.

FITZGERALD B, HARTNETT G and CONBOY K (2006) Customising agile methods to software practices at intel shannon. *European Journal of Information Systems, 15*(2), 200-213.

FITZGERALD B, RUSSO NL and STOLTERMAN E (2002) *Information systems development: Methods in action*. McGraw-Hill Education, New York, NY.

GIDDENS A (1984) The constitution of society: Outline of the theory ofstructuration. *Cambridge: Polity*.

GOH JM, GAO G and AGARWAL R (2011) Evolving work routines: Adaptive routinization of information technology in healthcare. *Information Systems Research, 22*(3), 565-585.

GREGOR S (2006) The nature of theory in information systems. *MIS Quarterly, 30*(3), 611-642.

HAMMER M and CHAMPY J (2009) *Reengineering the corporation: A manifesto for business revolution*. HarperCollins, New York, NY.

HARRIS ML, COLLINS RW and HEVNER AR (2009) Control of flexible software development under uncertainty. *Information Systems Research, 20*(3), 400-419.

HIRSCHHEIM R and KLEIN HK (1989) Four paradigms of information systems development. *Communications of the ACM, 32*(10), 1199-1216.

HUMMEL M (2014) State-of-the-art: A systematic literature review on agile information systems development. In *38th Hawaii International Conference on System Sciences (HICSS)*, pp. 4712-4721, Big Island, HI.

IIVARI J and MAANSAARI J (1998) The usage of systems development methods: Are we stuck to old practices? *Information and Software Technology, 40*(9), 501-510.

KIM E and RYOO S (2012) *Agile adoption story from nhn*. Paper presented at the Computer Software and Applications Conference Izmir, Turkey.

KLEIN KJ and KOZLOWSKI SWJ (2000) A multilevel approach to theory and research in organizations: Contextual, temporal, and emergent processes. In *Multilevel theory, research, and methods in organizations: Foundations, extensions, and new directions* (KLEIN KJ and KOZLOWSKI SWJ, Eds), pp. 3-90, Jossey-Bass, San Francisco, CA.

LANGLEY A (2007) Process thinking in strategic organization. *Strategic Organization, 5*(3), 271-282.

LATOUR B (1986) The powers of association. In *Power, actions and belief – a new sociology of knowledge* (LAW J, Ed.), pp. 264-280, Routledge & Kegan Paul, London, England.

LATOUR B (2005) *Reassembling the social: An introduction to actor-network-theory* (Vol. 1). Oxford University Press, Oxford, England.

LEE G and XIA W (2010) Toward agile: An integrated analysis of quantitative and qualitative field data on software development quality. *MIS Quarterly, 34*(1), 87-114.

LYYTINEN K, ROSE G and YOO Y (2010) Learning routines and disruptive technological change: Hyper-learning in seven software development organizations during internet adoption. *Information Technology & People, 23*(2), 165-192.

LYYTINEN K and ROSE GM (2006) Information system development agility as organizational learning. *European Journal of Information Systems, 15*(2), 183-199.

MANGALARAJ G, MAHAPATRA R and NERUR S (2009) Acceptance of software process innovations - the case of extreme programming. *European Journal of Information Systems, 18*(4), 344-354.

MARCH JG and SIMON HA (1958) *Organizations*. Wiley, New York, NY.

MARUPING LM, VENKATESH V and AGARWAL R (2009) A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research, 20*(3), 377-399.

MATHIASSEN L and VAINIO AM (2007) Dynamic capabilities in small software firms: A sense-and-respond approach. *IEEE Transactions on Engineering Management, 54*(3), 522-538.

MCLEOD L and MACDONELL SG (2011) Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys (CSUR), 43*(4), 24-56.

MOE NB, AURUM A and DYBÅ T (2012) Challenges of shared decision-making: A multiple case study of agile software development. *Information and software technology, 54*(8), 853.

MONTEALEGRE R and KEIL M (2000) De-escalating information technology projects: Lessons from the denver international airport. *MIS Quarterly, 24*(3), 417-447.

MÜLLER SD, MATHIASSEN L and BALSHØJ HH (2010) Software process improvement as organizational change: A metaphorical analysis of the literature. *Journal of Systems and Software, 83*(11), 2128-2146.

MYERS MD and YOUNG LW (1997) Hidden agendas, power and managerial assumptions in information systems development: An ethnographic study. *Information Technology & People, 10*(3), 224-240.

NERUR S, MAHAPATRA R and MANGALARAJ G (2005) Challenges of migrating to agile methodologies. [Article]. *Communications of the ACM, 48*(5), 73-78.

O'CONNOR CP (2011) *Anatomy and physiology of an agile transition*. Paper presented at the Agile Conference, Salt Lake City, UT.

ORLIKOWSKI WJ (1993) Case tools as organizational change: Investigating incremental and radical changes in systems development. *MIS Quarterly*, 309-340.

ORLIKOWSKI WJ (2000) Using technology and constituting structures: A practice lens for studying technology in organizations. *Organization Science, 11*(4), 404-428.

PASSOS OM, DIAS-NETO AC and DA SILVA BARRETO R (2012) Organizational culture and success in spi initiatives. *IEEE Software, 29*(3), 97-99.

PENTLAND B, HAEREM T and HILLISON DW (2009) Using workflow data to explore the structure of an organizational routine. In *Organizational routines: Advancing empirical research* (BECKER MC and LAZARIC N, Eds), pp. 47-67, Edward Elgar Publishing, Northampton, MA.

PENTLAND BT (1995) Grammatical models of organizational processes. *Organization Science, 6*(5), 541-556.

PENTLAND BT (2003) Sequential variety in work processes. *Organization Science, 14*(5), 528-540.

PENTLAND BT and FELDMAN MS (2005) Organizational routines as a unit of analysis. [Article]. *Industrial & Corporate Change, 14*(5), 793-815.

PENTLAND BT and FELDMAN MS (2007) Narrative networks: Patterns of technology and organization. [Article]. *Organization Science, 18*(5), 781-795.

PENTLAND BT and FELDMAN MS (2008) Designing routines: On the folly of designing artifacts, while hoping for patterns of action. *Information & Organization, 18*(4), 235-250.

PENTLAND BT, FELDMAN MS, BECKER MC and LIU P (2012) Dynamics of organizational routines: A generative model. [Article]. *Journal of Management Studies, 49*(8), 1484-1508.

PENTLAND BT, HAEREM T and HILLISON D (2011) The (n)ever-changing world: Stability and change in organizational routines. [Article]. *Organization Science, 22*(6), 1369-1383.

PENTLAND BT and RUETER HH (1994) Organizational routines as grammars of action. *Administrative Science Quarterly, 39*(3), 484-510.

PROJECT MANAGEMENT INSTITUTE (2013) *A guide to the project management body of knowledge: Pmbok® guide*. Project Management Institute.

ROBEY D, SMITH LA and VIJAYASARATHY LR (1993) Perceptions of conflict and success in information systems development projects. *Journal of Management Information Systems*, 123-139.

ROYCE WW (1970) Managing the development of large software systems. In *IEEE WESCON*, pp. 328-338, Los Angeles, CA.

RUMMLER GA and BRACHE AP (2012) *Improving performance: How to manage the white space on the organization chart*. John Wiley & Sons.

SALO O and ABRAHAMSSON P (2008) Agile methods in european embedded software development organisations: A survey on the actual use and usefulness of extreme programming and scrum. [Article]. *IET Software, 2*(1), 58-64.

SALVATO C (2009) The contribution of event-sequence analysis to the study of organizational routines. In *Organizational routines: Advancing empirical research* (BECKER MC and LAZARIC N, Eds), pp. 68-102, Edward Elgar Publishing, Northampton, MA.

SALVATO C and RERUP C (2011) Beyond collective entities: Multilevel research on organizational routines and capabilities. *Journal of Management, 37*(2), 468-490.

SARKER S, MUNSON CL, SARKER S and CHAKRABORTY S (2009) Assessing the relative contribution of the facets of agility to distributed systems development success: An analytic hierarchy process approach. [Article]. *European Journal of Information Systems, 18*(4), 285-299.

SARKER S and SARKER S (2009) Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research, 20*(3), 440-461.

SCHWABER K and BEEDLE M (2003) *Agile software development with scrum*. Prentice Hall, Upper Saddle River, NJ.

SCOTT WR (2008) *Institutions and organizations: Ideas and interests*. Sage Publications.

SRINIVASAN J, DOBRIN R and LUNDQVIST K (2009) *'State of the art' in using agile methods for embedded systems development*. Paper presented at the Computer Software and Applications Conference, Seattle, WA.

STRODE DE, HUFF SL, HOPE B and LINK S (2012) Coordination in co-located agile software development projects. *The Journal of Systems and Software, 85*(6), 1222-1238.

SUSCHECK CA and FORD R (2008) Jazz improvisation as a learning metaphor for the scrum software development methodology. *Software Process: Improvement and Practice, 13*(5), 439-450.

TEECE DJ, PISANO G and SHUEN A (1997) Dynamic capabilities and strategic management. *Strategic management journal, 18*(7), 509-533.

TRUEX D, BASKERVILLE R and TRAVIS J (2000) Amethodical systems development: The deferred meaning of systems development methods. *Accounting, management and information technologies, 10*(1), 53-79.

TURNER R and BOEHM B (2003) People factors in software management: Lessons from comparing agile and plan-driven methods. *CrossTalk Magazine, The Journal of Defense Software Engineering, 16*(12), 4-8.

VALACICH JS, GEORGE JF and HOFFER JA (2004) *Essentials of systems analysis and design*. Pearson Education, Upper Saddle River, NJ.

VERSIONONE.COM (2013) Seventh annual state of agile development survey. http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf, accessed June 16, 2014.

VIDGEN R and WANG X (2009) Coevolving systems and the organization of agile software development. *Information Systems Research, 20*(3), 355-376.

VOLKOFF O, STRONG DM and ELMES MB (2007) Technological embeddedness and organizational change. *Organization Science, 18*(5), 832-848.

WHITWORTH E and BIDDLE R (2007) *The social nature of agile teams*. Paper presented at the Agile Conference, Washington, DC.

WIKIPEDIA (2015) Systems development life cycle. http://en.wikipedia.org/wiki/Systems_development_life_cycle, accessed Jan 5, 2015.

XU P and RAMESH B (2007) Software process tailoring: An empirical investigation. *Journal of Management Information Systems, 24*(2), 293-328.