WHAT ARE PROJECT MEMBERS' ISD PROJECT MENTAL MODELS AND HOW DO THEY AFFECT THE MANAGEMENT OF ISD PROJECTS?

by

Yi-Te Chiu

A thesis submitted to the graduate program in Management in the School of Business

In conformity with the requirements for the degree of Doctor of Philosophy

Queen's University

Kingston, Ontario, Canada

April, 2014

Abstract

To help organizations better understand and improve the management of information systems development (ISD) projects, this dissertation aims to understand what ISD project knowledge and beliefs ISD professionals work with and how the knowledge and beliefs are organized in their minds. Drawing on the cognitive perspective using a mental model approach, I define a new construct, ISD Project (ISDP) mental model, which refers to ISD professionals' knowledge and belief structures that help them understand, conduct, and manage ISD projects. Particularly, two essential elements of ISDP mental models - content and structure – were explored. Regarding the content, forty fundamental concepts were derived from literature reviews and cognitive interviews with 19 ISD experts. Analysis of 95 ISD professionals' cognitive responses using Multidimensional Scaling revealed four types of evaluative beliefs - customer-, team-, enterprise-, and product-oriented beliefs. This new construct, along with its assessment procedures, provides a useful starting point for academics and organizations to explore the people factor in ISD.

To investigate the impact of ISDP mental models, I examined work relationships between project managers and developers where effective work relationships are crucial to project success. Specifically, I explored how the similarity of mental models and an understanding of others' mental models influence work relationships. Through a multiple case study on 6 project manager-developer pairs in different case conditions (i.e., similarity of mental models x accuracy of understanding), the results provide preliminary support that the project manager-developer pairs who hold accurate understanding have more effective implicit coordination - they are sensitive to one another's knowledge, beliefs, and preferences and they adjust their task and interpersonal coordination accordingly. Accurate understanding also stimulates the process of knowledge integration in which the dyad builds upon one another's knowledge to resolve project challenges.

This dissertation contributes to the literature on ISD project management by capturing the organization of ISDP knowledge in ISD professionals' minds and identifying underlying beliefs.

Furthermore, it contributes to an understanding of how project managers and developers can coordinate effectively when they have high cross-understanding, despite dissimilarities of knowledge and beliefs.

Acknowledgement

After over 2,000 days, I finally reach the finish line. It is not elation or exhaustion that hits me. Instead, I feel love and warmth surround me. There is a long list of people who have helped me to make the work possible. I would like to first thank the faculty and staff at Queen's School of Business who make QSB a wonderful research environment.

I would also like to express my sincere gratitude to members of my dissertation committee. Dr. Brent Gallupe pushed me to reconsider the essence of research and the relevance of my research to practitioners. Dr. Kathryn Brohman made me reflect on framing my research work and pointed out a wide variety of research opportunities. Dr. Jana Raver opened a window for me to see the world of shared cognition research and was a constant source of knowledge and inspiration. Dr. Deborah Armstrong and Dr. Patrick Martin generously provided me with thought-provoking advices and warm support. I especially want to thank my supervisor and mentor, Dr. Sandy Staples, for always being there for me emotionally and physically and for showing unwavering faith in me when I had self-doubt. This work would never be possible without him. He deserves many thanks which I cannot describe in simple words.

I would like to also thank Dr. James McKeen, Dr. Jane Webster, and Dr. Yolande Chan. Each of you has a profound influence on me and helps me build the character as a scholar. I must acknowledge my friends and colleagues from my PhD cohort – William Cram, Eruani Zainuddin, Paola Gonzalez, Yi Zou, Nadège Levallet, and Suchit Ahuja. They provided me with intellectual challenges and amazing support. A special thanks to Jacqueline Corbett. She was always ahead of me but often checked if I am coming and offered invaluable advice and help. Acknowledgement also goes to people who participated in this dissertation. Although I cannot list your names for reasons of confidentiality, your insights and experience make significant contribution to the findings. There are too many names for me to mention. I am grateful to each of my friends who stood by me through my PhD.

Finally, I owe my deepest gratitude to my parents and my older brother. Their encouragement and support never falter. This dissertation is dedicated to my wife, Ya-ting Lin, who gave up so much for me to complete this work.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Problems

The importance of information systems (IS) has dramatically grown over the past few decades as shown in the significant organizational spending on IT (4.3% of revenue, Gartner Research, 2010). A substantial number of organizations depend upon IS to improve their business efficiency and effectiveness, while expecting to gain competitive advantages from their IS investment. Despite the importance of IS, many organizations suffer from high failure rates in their IS projects (Standish Group, 2009), which is costly (e.g., 78 billion dollars per year in U.S.). The difficulty in the implementation of IS largely stems from the complexity of information systems development (ISD), embodying not only the management of multiple technical components but also the management of people and change (Xia & Lee, 2003).

A burgeoning number of ISD methodologies have illustrated diverse mindsets to cope with complexity. Generally speaking, the methodologies, defined as "a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management, and training for developers of information systems" (Maddison et al., 1984, p. 418), can be differentiated into two paradigms: Process (e.g., Waterfall, CMMI, and PMBOK) and People (e.g., Agile). The proponents of the former argue that careful upfront design, meticulous top-down planning, and rigorous quality control reduce the risk of a runaway project. Advocates of the later emphasize a need to capitalize on each individual's strength, including both development teams and customers, to deal with highly volatile, ambiguous, and uncertain requirements. While both paradigms demonstrate some successful anecdotal stories, in reality, the efforts in doctrinally implementing work practices underlying each do not often come to fruition for organizations. These challenges include resistance to these methodologies (Chow & Cao, 2008), absence of repeatable benefits (Avison & Fitzgerald, 2003), a high project failure rate (Standish Group, 2009), and growing concerns over ineffective project delivery processes (Gartner, 2013). Clearly, there is no one-size-fits-all solution. Practitioners and researchers have suggested a need to tailor methodologies to different organizational cultures and project characteristics, such as project size, developers' knowledge and skills, the degree of novelty , and the type of application (Cockburn, 2005; Keith, Demirkan, & Goul, 2013). Fitzgerald et al. (2002) call for more research on the methodology-in-action[1], which is the methodology adapted and used in organizations.

As opposed to prior attention to methodologies and methodology-in-action, I contend that that more focus must be paid to the *people* component, specifically ISD professionals' knowledge and beliefs about project management. In this dissertation, I refer to this concept as the ISD project (ISDP) mental model[2]. The ISDP mental model is composed of fundamental concepts[3] that people abstract from their understanding of ISDP management and represents a systematic arrangement of these concepts. The mental model is important because it provides a snapshot of one's knowledge and unearths one's belief , which can predict how a project member will likely behave in ISDPs (Kelly, 1955).

A comprehensive understanding of mental models in the ISDP context may resolve certain challenges that currently hinder the implantation of ISDPs (see starbursts in Figure 1). First, it is

---

[1] The precise term that has been used in literature is method-in-action (B. Fitzgerald, Russo, & Stolterman, 2002). While *Methodology* and *method* are often used interchangeable, there is continued debate on their differences. Methodology is generally a more inclusive term and encompasses philosophies and guiding principles, whereas method generally refers to specific procedures and processes in IS development (Iivari, Hirschheim, & Klein, 2000). We chose methodology in this paper because of its prevalence in the practitioner community and because the "guiding principles" component is essential to our study.

[2] To set up a boundary for the concepts proposed in this study, the ISDPs referred to here are limited to those projects aimed at creating information systems that meet customers' needs and encompass primary project phases (initiation, design, execution, monitoring and controlling, and closing). Therefore, projects related to ISD operations and maintenance are not included in this study. Additionally, ISD professionals are referred to as project members who engage in the above developmental activities. Their job titles are as diverse as project manager, designer, analyst, programmer, tester, and other relevant project roles.

[3] Concept is a meaningful unit in memory.

essentially people that make methodologies effective. Mental model provides a lens to understand how ISD professionals comprehend methodologies and how they tailor and respond to methodology-in-action. ISD professionals can have very different understandings of methodologies; for instance, one survey shows that participants proposed over one hundred different definitions for Agile methodologies (Voke, 2012). Until employees' mental models are examined, it would be challenging to know whether organizations have a repertoire of expertise to tailor methodology-in-action or whether espoused methodology-in-action fits individual project members. Second, ISD methodology is not a guaranteed solution for ISDP management. ISD methodology, regardless of its comprehensiveness, cannot completely inform ISD professionals regarding what must be accomplished and how to approach every incident arising over the course of a project. Eventually, it is people who comprehend the project environment and make decisions in response to various project challenges. Their ineffective mental models cause poor judgment on project challenges (Highhouse, 2001). Third, project success lies in collaboration. Collaboration between project members leads to tangible deliverables, suitable decisions, and innovative ideas. While divergent mental models can enable a team to respond appropriately to a volatile project environment, differences can also create conflict (Jehn, Northcraft, & Neale, 1999; Liang, Jiang, Klein, & Liu, 2010). Project managers can readily manage diverse behaviors if they understand their teams differences in knowledge and beliefs (Clark, Feldon, van Merriënboer, Yates, & Early, 2008; Kim, Cable, & Kim, 2005).

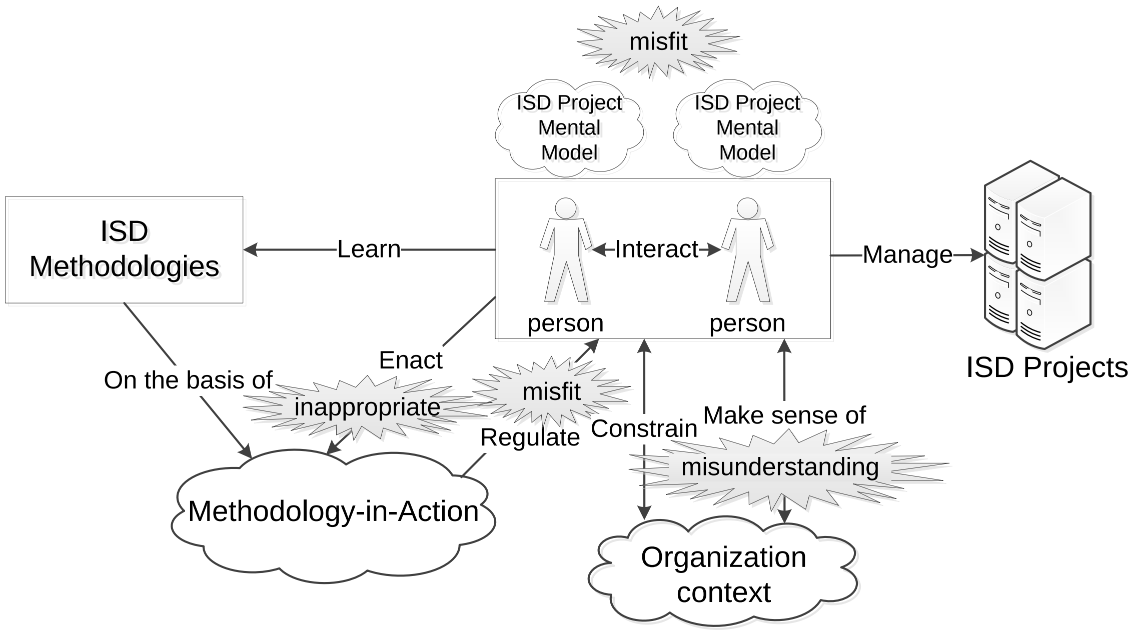


Figure 1 A Framework for Research on ISDP Mental Models[4]

The objectives of this dissertation are twofold. First, I will investigate and define what a project member's ISDP mental model is. Second, drawing on the ISDP mental model construct developed previously, I will explore how people who have diverse ISDP mental models can work together effectively.

1.2 Topic Overview and Research Questions

To understand ISD professionals' knowledge and beliefs about ISD project management, I take a cognition perspective. Cognition refers to the mental processes that take place in gaining and understanding the information of others and the social situation (Groome, 2006). It underpins an individual's thoughts, feelings, and behaviors (Hamilton, 2005). Cognition guides one's attention to specific aspects of a project and can predict one's decision-making and behaviors (Rouse, Cannon-Bowers, & Salas, 1992). Specifically, I draw on a mental model approach because a mental model, an

[4] The diagram is adpated from Fitzgerald et al. (2002).

individual's organized knowledge and belief structure (Cannon-Bowers & Salas, 2001; Rouse et al., 1992), serves an important function in the process of cognition. More discussion about cognition and mental models will be explicated in Chapter 2.

Mental models have particular value in complex tasks, such as military combat teams (Lim & Klein, 2006), air traffic control (Smith-Jentsch, Mathieu, & Kraiger, 2005), nuclear plant operations (Waller, Gupta, & Giambatista, 2004), and strategic investment issues in banks (Hodgkinson, Bown, Maule, Glaister, & Pearman, 1999). Complex tasks demand individuals and teams respond to novel and equivocal issues, which requires individuals to acquire relevant knowledge and share it with other project members, thereby integrating what each other knows to resolve a problem. Without appropriate mental models, individuals would have difficulty in understanding problems and absorbing new knowledge, let alone further discussion with other team members. Although individuals are allowed to learn from errors and adjust strategies in response to errors over the course of a project, particularly when dealing with less critical tasks, overt and lengthy coordination takes its toll on the effectiveness and efficiency of team outcomes (Entin & Serfaty, 1999; Yuan, Zhang, Chen, Vogel, & Chu, 2009).

ISD tasks inherently require collective, complex, and creative efforts, and thereby warrant a consideration of mental models. Imagine the efforts needed to understand users' needs, then to shape these needs into technical design, delegate this work to developers, coordinate with developers/users/other stakeholders, and finally, integrate and implement the systems. Information systems are indeed the synthesis of stakeholders' understanding and knowledge, which are abstract and constantly in the state of flux. Coordinating people who have different knowledge is undoubtedly a daunting task. The market pressure in a competitive environment, which leaves little margin for mistakes and inefficiencies, further compounds this complexity. Moreover, global collaboration faces the difficulty of coordinating project members in different locations and time zones (Espinosa, Slaughter, Kraut, & Herbsleb, 2007). The complexity of ISDPs, therefore, calls for a more in-depth analysis of knowledge and beliefs, which can be found in mental models.

IS research on mental models is still emerging. Existing IS literature can be distinguished into four themes: identification of the content of mental models, accuracy of mental models, similarity of mental models, and shift in mental models. Identification of the content aims to capture the meanings of concepts – a meaningful unit in memory - embedded in mental models related to IS problem domains, that is, what constitutes a mental model. This type of research serves as the foundation of the mental model research. There is no way to assess accuracy, similarity, and shift of mental models without an appropriate definition of mental models. For instance, Nelson et al. (2000) elicited thirty-three concepts and their relationships about software operations support using interviews and a causal mapping technique.

Accuracy assists ISD professionals in developing more accurate mental models, which helps achieve improved performance in IS activities such as programing (Cooke & Schvaneveldt, 1988; W. W. F. Lau & Yuen, 2010; Shaft, Albert, & Jasperson, 2008), conceptual modeling (Keng Siau & Tan, 2005a), database manipulation (Keng Siau & Tan, 2006), and requirement determination (Kudikyala & Vaughn, 2005). The degree of accuracy is assessed by comparing individual mental model to what is believed to be the most accurate, such as experts' mental models.

Similarity of mental models examines consistency, convergence, and compatibility of mental models among actors or between actors and tasks. Compatibility between actors refers to the overlap of mental models rather than the presenting with identical mental models. Actors with similar mental models are more likely to work together effectively because of their comparable knowledge and beliefs, which will result in better communication and coordination (Mohammed, Ferzandi, & Hamilton, 2010). Actors with mental models that are compatible regarding the knowledge content (both task-specific and task-related) that is required to perform a specific task are more likely to accurately comprehend problems and effectively conduct tasks. For instance, Shaft and Vessey (2006) found that when developers' mental models of the software (i.e., technology, domain, and application knowledge for the software) were compatible with knowledge needed for software modification tasks, their task performance improved.

Finally, *shift in mental models* concerns changes in existing mental models and how such changes affect ISD activities. This is illustrated by a study that investigated the consequences of a shift in technological frames—another term for mental models—on determining requirements (E. J. Davidson, 2002). Four dimensions of technological frames were discovered in the context of IT: business values, delivery strategy, usage, and design and capabilities. Shifts in these frames altered project members' understanding of requirements, thereby, impacting how they determined requirements and project deliverables.

Notwithstanding the value of a mental model perspective, little is known about what mental models practitioners apply to address complex ISDP management issues. What are the domain-specific concepts they adopt in the management of ISD projects? How do these concepts link and affect project members' thinking and behavior? A lack of defining project members' mental models hinders the further understanding of appropriate mental models to address looming engineering and management issues in ISD projects (the theme of accuracy). It also inhibits the understanding of potential synergies and conflicts among project members who have different mental models (the theme of similarity). To this end, this dissertation proposes a new construct, ISDP mental model, and defines it as project members' knowledge and belief structures that help them to understand, conduct, and manage ISD projects. This leads to the first research question:

Research Question 1: *What is a project member's ISDP mental model? What are the core ISDP concepts that comprise a project member's knowledge and belief structures about the management of ISD projects and how are these concepts interrelated? (Study 1)*

To further understand how knowledge and belief structures affect ISD projects, I examine the interplay of mental models, particularly the similarity/diversity of mental models and an understanding of each other's mental models, between project managers and developers. From the similarity/diversity perspective, mental models can be considered as deep-level diversity, namely differences in knowledge, attitudes, beliefs, and values (Harrison, Price, & Bell, 1998). Deep-level diversity engenders potential task and relationship conflicts (Jehn, Chadwick, & Thatcher, 1997) and hampers communication, predictability, attraction, and trust (J. R. Edwards & Cable, 2009). Despite the substantial research on the deep-level diversity, there is a continued need, from both the theoretical and practical consideration, to understand how diverse knowledge and beliefs can be capitalized on while at the same time avoiding potential conflicts.

To address this need, I investigate the relationships between diversity (i.e., diverse mental models), understanding, and their impacts on work relationships. I argue that understanding others' mental models should play an essential role because it has been an essential process for interpersonal interaction. Both Allport's (1954) Contact Hypothesis and research on team cognition (see recent reviews in Cooke et al. (2007)) imply that the process of understanding reduces interpersonal biases and improves work interaction even when diversity exists. For instance, the team mental model (TMM), one of the popular constructs in shared cognition research, suggests that when team members build similar mental representations of taskwork and teamwork, the converged mental representations improve team processes, including interpersonal outcomes such as better communication and coordination (see recent review in Mohammed et al., 2010). The TMM construct implies a need for an understanding of tasks and team members. However, it does not separate understanding between each other (i.e., the process of understanding, degree of understanding, and what should be understood) from the component of similarity (i.e., when individual team members' mental models converge and what a collective mental model should look like). In other words, an interest of TMM is in the impact of a converged team mental model on team processes and outcomes rather than the process of TMM development. Arguably, there is a need to precisely articulate the concept of understanding. This study focuses on the concept of understanding, responding to Huber and Lewis's (2010) recent call for research on cross-understanding, "an accurate understanding of the mental models of other members" (Huber & Lewis, 2010).

Following this line of argument, study 2 aims to explore how an understanding of others' mental models affects work relationships, especially when the project managers and developers do not possess similar mental models. Since the linkage between mental models and work relationships is still ill-defined, this study involves an exploratory search for the impacts of understanding and similarity on different dimensions of work relationships, including attitudes (e.g., trust, loyalty, and respect) (Ferris et al., 2009) and interaction patterns between the dyad, such as helping behaviors (Halbesleben, 2011) and effective coordination (Gittell, 2011). I seek to build a preliminary theoretical model for the work relationships between project managers and developers with an emphasis on cognition-focused constructs (i.e., similar ISDP mental models and an accurate understanding of mental models). The findings have potential to contribute to the literature on ISD project management in particular and research on work relationships and cognition in general. This leads to the following broad research question and three sub-questions.

Research Question 2: *How does the interplay of ISDP mental models affect work relationships between project managers and developers?*

(a) How does the similarity of ISDP mental models between project managers and developers affect work relationships?

(2) How does an understanding of others' ISDP mental models affect work relationships?

(3) What are the major drivers of an accurate understanding of others' ISDP mental models?

1.3 Motivation and Potential Contributions

Being an ISD professional for over a decade, I feel that ISD is not pure engineering and rather should be something highly human centric. Although there have always been sophisticated engineering approaches laying down techniques, standards, and procedures for projects (as they are advocated by organizations and consultants), it turns out, in real life people explain and weigh these approaches differently. Sometimes our co-workers may speak very different vocabularies and convey ideas that we never thought of. We may not be able to proceed for the lack of understanding or we simply may not care about what they are expressing. Until we can build a common understanding of one another, guessing and ineffective communication spread all over the course of projects. I saw how misunderstanding has created conflicts and toxic work environments; for instance, my co-workers attributed our project managers' tight control as a gesture of distrust or a lack of knowledge about programming (many programmers believe programming is an art and should be developed in a highly creative environment). In this dissertation, I hope to help practitioners gain a better understanding of themselves and their co-workers. That is to say, we and other project members may never be the same in terms of knowledge and beliefs; but it is important that we have a solid and accurate understanding of each other's knowledge and beliefs. The target is to reduce losses during our interaction and enable better collaboration outcomes. Ultimately, better collaboration leads to a resolution of complex issues in ISD projects.

I believe that this dissertation can shed new light on the human-centric aspect of ISD. Study 1 answers the question of what constitutes a mental model regarding the management of ISDPs. In developing a new ISDP mental model construct, I make a major contribution by synthesizing the body of knowledge related to ISDPs, including both project management and software engineering, combining this with concepts from cognitive science, and clarifying how complex knowledge about ISD projects is understood and organized by ISD professionals. Additionally, I investigate belief systems embedded in mental models. Belief systems indicate why ISD professionals learn and apply certain concepts. By capturing project members' belief systems, we are in a better position to predict project members' interactions to ISDPs. I believe that findings from this dissertation not only improve our understanding of knowledge about ISDP management but also the understanding of people working within ISDPs. This in turn can result in more effective management of ISDPs.

Study 2 addresses the issues of diverse mental models between project managers and developers. I suggest understanding as a key component in intensive and complex work interaction processes and explore the impacts of understanding on work relationships, specifically coordination of ISD tasks between project managers and developers. In pursuing this, I contribute to the literature by developing a cognitive perspective to understand how individuals in dyads processes their understanding of one another and contributes to effective collaboration in projects. In other words, the theoretic framework I developed is the first one to understand the diversity issue in the ISD context by looking into accurate understanding of mental models. This provides organizations and researchers a new way to look at many diversity issues in the ISD context, such as different mental models between partners in outsourcing/offshoring projects or different mental models between business managers and IT managers. The cognitive perspective using a mental model approach would be of value to recognize the causes of unproductive collaboration behaviors and find out potential interventions to address a lack of understanding between diverse mindsets. I hope that the readers can not only gain insights into mental models, knowledge about ISD projects, diversity of mental models, and understanding of mental models, but also apply the perspective to their area of interests.

1.4 Terminology

- ISD projects: projects that aim to create information systems that meet customers' needs and have been through project phases from initiation, design, execution, monitoring and controlling, and closing.
- ISD professionals: people who engage in the development activities of ISD projects (defined previously). They are project members of ISD projects and their roles can be project managers, designers, analysts, programmers, testers, and other similar project roles.
- Cross-understanding: the extent to which the dyadic partners (project manager and developer in this study) possess an accurate understanding of the mental models of other members (Huber & Lewis, 2010).
- Cause-effect belief: how strongly a concept is believed to influence another concept (Chattopadhyay, Glick, Miller, & Huber, 1999).
- Evaluative belief: the desired properties or states that the individual prefers, expects, or demands (also known as utilities, values, and preferences) (Huber & Lewis, 2010).
- Concept: a meaningful unit in memory systems.
- Mental model: an individual's organized knowledge and belief structure, which helps one to explain, understand, and predict events occurring in the environment (Cannon-Bowers & Salas, 2001; Rouse et al., 1992).
- Information systems development project (ISDP) mental model: ISD professionals' knowledge and belief structures that help them understand, conduct, and manage ISD projects.

1.5 Outline of Dissertation

The dissertation is structured as follows. The fundamental literature on mental models is reviewed and reflected in Chapter Two. Chapter Three defines and examines a new construct – ISDP mental models. Chapter Four extrapolates the work relationships between project managers and developers in light of the similarity and understanding of ISDP mental models. Grounded by data, I introduce a theoretical model for the work relationships between project managers and developers. Finally, Chapter Five summarizes research contributions and proposes future research directions.

Chapter 2

Foundation Literature

Before delving into two studies, an explication of mental models is needed to clarify how I am going to apply the concept and where the gap in IS research on mental models is. Chapter Two is devoted to providing a fundamental literature review on cognition and mental models. This chapter began with establishing some basic nomenclature by describing the cognitive processes. Then I indicated that the mental model is associated with the long-term memory, which can be further divided into three types of memory (semantic, episodic, and procedural). Among the three types of memory, I highlighted the semantic memory because it underpins mental models. Then, I briefly introduced four views on the representation of knowledge in mental models (classic, prototype, exemplar, and theory). I suggested that the theory view is appropriate for understanding ISDP mental models (Section 2.2). After that, in Section 2.3, I classified different types of mental models that are relevant to ISD projects and suggested how the ISDP mental model might fit in the classification of mental models for ISD projects. I concluded the chapter by summarizing potential gaps. The literature specifically related to study 1 and study 2 will be addressed in Chapter Three and Chapter Four respectively.

2.1 Cognition and Mental Models

According to the information processing perspective (Schroder, Driver, & Streufert, 1967), cognition can be conceptually analyzed at different stages, including attention, encoding, elaboration, mental representations, and retrieval, as illustrated in Figure 2. Given the limited capacity in human working memories (G. A. Miller, 1956), people selectively acquire information based on the importance and salience of cues in adapting to a stimulus-rich environment and to avoid cognitive overloading. The information is then interpreted, elaborated, and finally encoded in the long-term memory. Depending on types of information, information can be encoded and stored into one or more memory subsystems: episodic memory, semantic memory, and procedural memory (Tulving, 1985). Episodic memory contains specific events one has experienced including when and where they occurred; Semantic memory contains general knowledge not associated with a particular time and context but rather facts, meanings, concepts and knowledge that a person has acquired about the external world; Procedural memory is the unconscious memory of actions, skills, and operations (e.g., typing, riding a bike), which is relatively automatic and habitual (Reed, 2007).

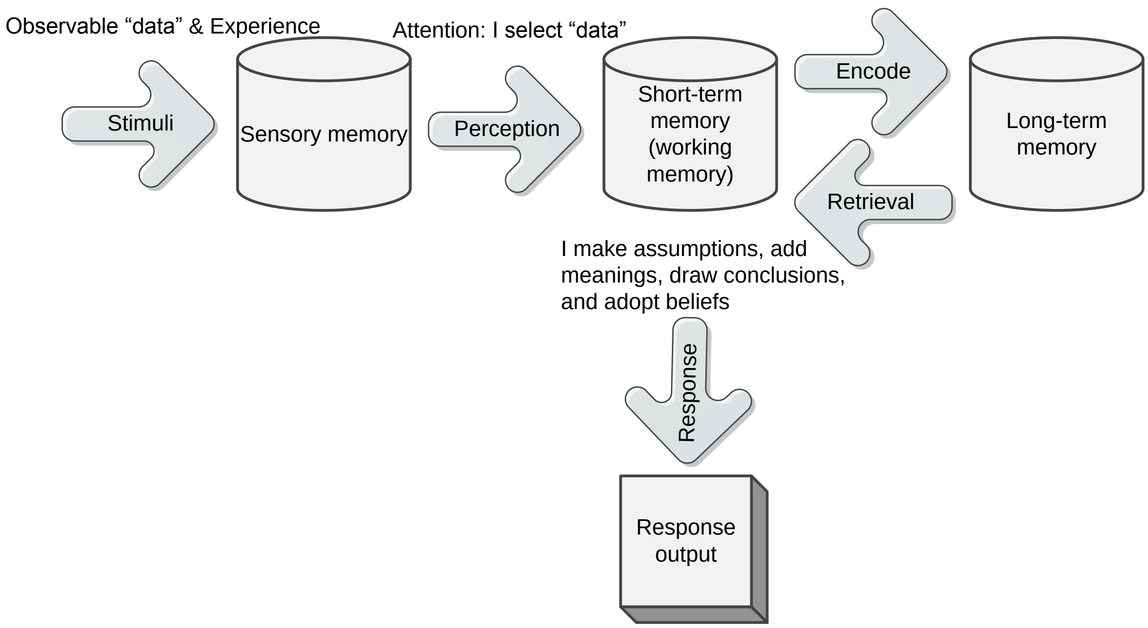


Figure 2 Information Processing Perspective and Memory

ISDP activities involve all three memories. For instance, a project team sets up a system architecture that supports automatic regression testing. Once automatic regression testing is implemented, all team members benefit from the defect control's efficiency because regression testing can now be run automatically and consistently. The incident is stored in each team member’s episodic memory. Given individuals' diverse experience and cognitive processing, episodic memory between individual team members varies. For example, a person who implements automatic regression testing architecture would know specific details, such as how to integrate a source control system with automated testing tools, while other team members may simply know how to use it. Such experience will likely become part of all of the team members' procedural memory. Depending on their knowledge and beliefs, people may derive new concepts from the experience as they recognize the importance of test-driven development (or defect prevention) to respond to constant changes and ensure software quality. The new concept - test-driven development - is therefore summarized and encoded in semantic memory.

This dissertation focuses on the organized knowledge based on semantic memory, which individuals use to make sense of information (e.g., seek causes of behaviors and events), decide what to do, and predict what will happen (Collins & Quillian, 1969; Mathieu, Heffner, Goodwin, Salas, & Cannon-Bowers, 2000). A variety of terms have been used to describe the organized knowledge, such as mental models, mental representations, cognitive structures, cognitive maps, knowledge structures, and schemas (Schuelke et al., 2009). It should be noted that mental model and semantic memory are highly related but have been used differently. Both are a hypothetic structure used to depict how human's knowledge and information are stored and structured. However, the nuance between the two concepts is that semantic memory is usually considered a repository and primarily used descriptively in cognitive science research whereas mental model, an individual's organized knowledge and belief structure (Mohammed, Klimoski, & Rentsch, 2000; Rouse & Morris, 1986), is contextually dependent and application-oriented. Depending on the domains, people can develop multiple mental models and use them to explain, understand, and predict events occurring in the environment. The term mental model has been primarily used prescriptively in management and industrial psychology. Given the purpose of this dissertation, mental model is the focus but the introduction of memory in this Chapter provides the scientific underpinnings.

Technically speaking, mental models must be comprised of content and structure (Walsh, 1995). Content is the knowledge of a given concept, such as test-driven development, whereas structure is the systematic arrangement of concepts, such as the relationship between test-driven development and continuous integration of system builds. Considering cognitive processes, As people encounter new events, they retrieve concepts and memories of related experiences from mental models and imbued with further insights develop a new understanding that may influence their behavior (Hamilton, 2005). For instance, returning to the above scenario, team members who have formed the test-driven development concept may propose the adoption of test-driven development in their future projects; the concept is now part of their mental model. Therefore, when facing similar issues, people are able to reason from these concepts to make a decision.

However, not all cognitive activities entail such detailed and effortful information processing. Depending on the characteristics of issues and individual differences, information processing can be automatic, involving the development and deployment of heuristics. The intuitive decision-making does not entail detailed information processing that is depicted in Figure 2 but rather relies on information in memory and makes a judgment without deeper elaboration. Given the complexities of projects and associated tasks, I assert that deliberate thinking would dominate the cognitive processes of ISD professionals. That is, ISD professionals are more likely to process ISD relevant information carefully. Nonetheless, once decision or development tasks turn into a daily routine, ISD professionals may switch back and forth between these two modes of information processing – deliberate and automatic. No matter which mode dominates, mental models that store information structurally in memory, which represents core beliefs and knowledge about a phenomenon, should play a critical role and dictate thinking and actions. This study is dedicated to investigating the mental models of ISD professionals with an emphasis on deliberate thinking.

2.2 Structure of Mental Models

There are abundant theories regarding how concepts are represented in mental models and how the organization of concepts helps people retrieve information more efficiently and effectively. Kunda (1999) summarized four popular representations. It should be noted that I am not attempting at an integration of different views; instead I am applying well-accepted features from these views to understand how they are related to different phases of cognitive processing in an ISD context. Then, I choose the perspective that is most suitable for the purpose of this dissertation.

First, the classic view of mental models, one of the earlier theories, argues that each concept should have a required set of attributes and different concepts should be explicitly distinct in terms of attributes. The idea of common attributes and characteristics across different concepts provide a foundation of mental models. However, the view is characterized by its strict definitions of each concept. Concepts are categorized when they fully meet the definitions. The view resembles academics' attempt to categorize knowledge (see Iivari et al.'s (2000) taxonomy of ISD methodologies). In reality, this view is less likely to represent one's mental model. Even for ISD experts, they may not have such a clear-cut mental model.

Second, the prototype view (or the probabilistic view) of mental models is fuzzier, namely there is no strict definition of a concept, compared to the classic view. Concepts, albeit with different attributes, can be further grouped together as a prototype if they share similar characteristics with a prototype. That is, each concept does not have a clear-cut boundary but is organized according to a generic family resemblance. For instance, one of the prototypical features of the concept "defect prevention" is the root cause identification of previous defects in order to prevent recurring issues. Concepts sharing the similar characteristics, such as test-driven development, automated testing, collection of statistics on sources, and adoption of design, can be categorized into the same group (or prototype) under this view.

Third, the exemplar view of mental models posits that rather than general prototypical concepts, it is a set of exemplars that is represented. In this view, concepts are not formed independently. Rather, the mental model is comprised of different incidents that people experience. This view has been criticized for ignoring abstract concepts and letting any instance being a concept (Medin, 1989). Currently, researchers believe that both the exemplars and abstract concepts should co-exist as people may reason based on individual exemplars if they do not form abstract concepts.

The above three views are primarily similarity-based. These theoretical views have been tested through categorization tasks but many of them were done in laboratory settings with classification as the sole purpose (Markman & Ross, 2003). However, they may not be applicable to a natural decision-making setting because they miss one important aspect of knowledge – the theoretical relationships among concepts (F. C. Keil, 1992). Namely, the fourth view – the theory-based view – suggests that concepts should be grouped together due to theoretical reasons rather than the atheoretical similarities of attributes and properties. The "theorization" process is more likely to occur in our daily activities, such as communication, predictive inference, or preference formation. For example, the "workforce planning" concept is about constituting a team based on complementary expertise and activities; the "capability development" concept is about the provision of training and mentoring. Individuals learn and associate these two concepts because they both aim to cultivate human capital and IT capabilities (Bharadwaj, 2000).

Theoretical relationships between concepts are consistent with goal-derived category literature that argues people learn how and why certain salient concepts interrelate in light of goals. Goals guide how specific concepts are given attention, encoded, and organized (Barsalou, 1991). In an ISD context, practitioners, consciously or unconsciously, pursue these goals to meet needs for personal advancement and growth, peer recognition, and work requirements (Haslam, Powell, & Turner, 2000). Over time, the concepts are formed and associated. The application of certain concepts becomes habitual and certain concepts become chronically accessible altogether (Fishbach & Ferguson, 1996).

Given the importance of the theory-based view in learning and natural decision-making settings, in study 1 (Chapter Three), I examine the structure based on the goal perspective. On the other hand, I still take into consideration the notions of attributes, properties, and examples of concepts coming from the other three views and apply them when I derived concepts using mental model elicitation techniques (I coded attributes and properties of ISDP concepts from ISD methodologies. Furthermore, examples of concepts mentioned by my interviewees were categorized into suitable ISDP concepts. See Section 3.3 for the discussion of different elicitation techniques).

2.3 Classification of Mental Models in ISD

People possess multiple mental models that respond to different tasks. To differentiate between ISDP and non-ISDP mental models, we propose the following taxonomy. Drawing upon established classifications of mental models and IS Body of Knowledge (Cannon-Bowers & Salas, 2001; Iivari, Hirschheim, & Klein, 2004), I divided ISDP mental models into four first-level categories*: task-specific knowledge, task-related knowledge, knowledge of teammates*, and *beliefs*. Figure 3 shows the relationships between these four categories and IS Body of Knowledge. They will be further discussed in the following sections.

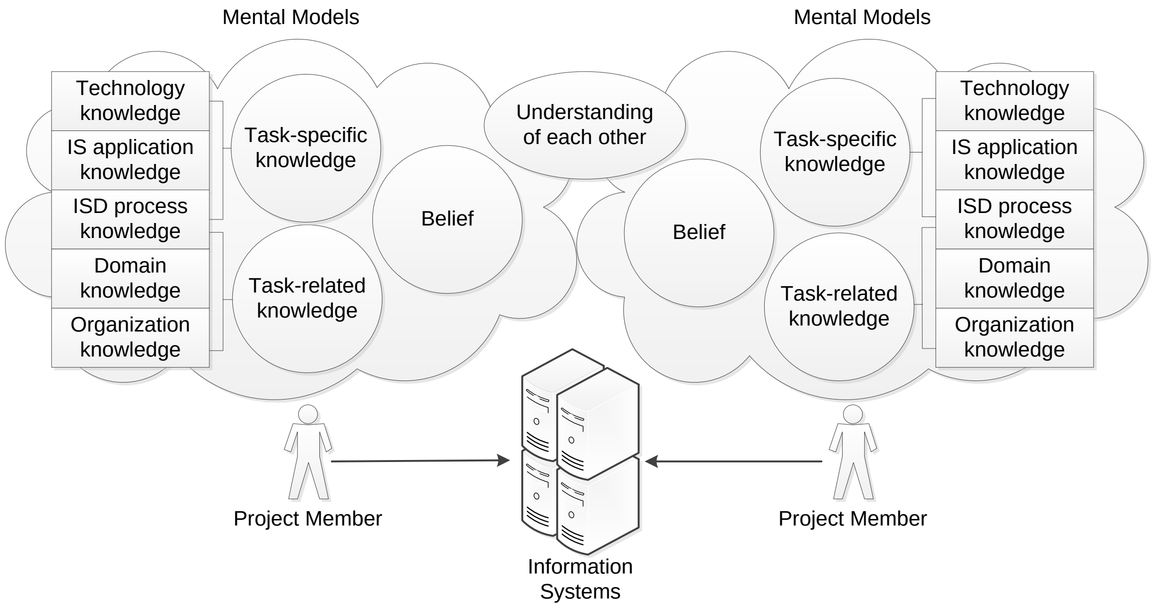


Figure 3 Types of Mental Models in ISD

Task-specific Knowledge. Task-specific knowledge is unique to ISD tasks. It pertains to the core knowledge in a particular domain and includes facts, procedures, and actions necessary for a given task, such as specific knowledge about system analysis and design. Task-specific knowledge is comprised of technology knowledge, IS application knowledge, and ISD process knowledge (Iivari et al., 2004).

Technology knowledge refers to the knowledge about specific technological components, e.g., operation systems, programming languages, database management systems, networking, hardware, and other application tools. For instance, Armstrong and Narayanan' (2005) mental representations of object-oriented programming, using revealed causal mapping, revealed technology knowledge specific to programming. They identified 19 object-oriented concepts and showed the relative importance of concepts in programmers' mind.

IS application knowledge refers to the knowledge about IS applications, including their infrastructure, functionalities, and use. For example, Tan et al. (2009) applied Repertory Grid Technique, one of the mental model elicitation techniques (see Section 3.3 for greater details), to understand web designers' knowledge about the good design of B2C websites.

ISD process knowledge is concerned with the performance (i.e., design, analysis, and implementation) and management (i.e., planning, evaluating, and regulating) of ISD (Andersen et al., 1990). There is no doubt that knowledge about performing ISD is task-specific knowledge, such as requirement construction and architectural design. For example, Kudikyala and Vaughn (2005) captured both developers' and users' mental models on business requirements and compared differences between developers and users.

Different from the performance of ISD, knowledge about the management of ISD can be categorized as either the task-specific or task-related knowledge. This multifaceted nature can be traced back to different contributors in the management campaign and the engineering campaign. For instance, some knowledge, advocated by engineers, is very specific to ISD projects, such as COnstructive COst MOdel II (COCOMO II) to estimate the cost, effort, and schedule, while others, such as knowledge about project monitoring and tracking, are general enough to apply in different settings. There is little attention being paid to mental models about the management of ISD. Prior studies touched upon project members' mental model on general issues for the management of ISD but rarely got into how they apply their knowledge to resolve issues.

For instance, Moynihan (1996; 1997) studied developers' mental models on contextual factors of ISD projects (i.e., characteristics of customers and applications) and identified 21 concepts critical to projects, such as "Who we will be working through: users versus the IT department, individuals versus committees" and "Need to integrate/interface with other systems". In a similar vein, Pankratz and Loebbecke (2011) identified success factors of IS projects from project managers' mind. Their results contribute to our knowledge about the issues project managers and members may pay attention to but not for the management of ISD projects per se. Lee and Truex (2000) pushed this stream of research toward the mental model about the management of ISD but unfortunately not in depth. By studying students who enrolled in an ISD methodology course, they found that students hold stronger methodological stance (i.e., pro-methodology) over the duration of the course. Their findings contribute to the potential for intervention in mental shifts, but not to a better understanding of how ISD professionals apply ISD process knowledge to their projects.

Task-related Knowledge. Task-related knowledge, in contrast, is knowledge used to accomplish ISD tasks, but is also useful for other tasks. Generally speaking, it can be applied across different problem domains. Some types of ISD process knowledge, particularly the management aspect, fall under task-related knowledge, as do application domain and organization knowledge. The management aspect of ISD process knowledge is capable of being both specific to ISD projects, such as Constructive Cost Model II (COCOMO II), or general enough to apply to different settings, such as in project monitoring and tracking. Domain knowledge is the knowledge related to a specific application where systems are built: for example, warfare knowledge is needed to build a combat system (B. Curtis, Krasner, & Iscoe, 1988). Organization knowledge refers to the knowledge of social and economic factors that are embedded within organizational processes, which may affect ISDPs. Ackerman and Eden (2005) laid out a causal map, a type of mental representations, of the management of complex projects in the aerospace, transport systems, shipbuilding, and civil engineering by interviewing project participants using group support systems. While not directly related ISD projects, this example illustrates the importance of organization knowledge in complex projects.

Knowledge of Co-workers. ISDPs are never realized by one individual and require close collaboration; therefore, a critical component of mental models is the understanding of project members' knowledge, skills, abilities, and preferences. Past research has demonstrated that an awareness of other team members' knowledge improves the use of the information (Hsu, Chang, Klein, & Jiang, 2011) and team performance (He, Butler, & King, 2007; Hsu et al., 2011).

Belief. Belief is another vital component of mental models. The boundary between beliefs and knowledge is vague as knowledge is also called justified true beliefs (Moser, 1995). To avoid ambiguity, the beliefs I discuss in this dissertation are evaluative beliefs, define as "what is wanted, what is best, what is desirable or preferable, what ought to be done" (Scheibe, 1970). Evaluative beliefs differ from the three types of knowledge described above, in that they are characterized by "descriptive states of nature that one knows to be true" (Mohammed, Klimoski, & Rentsch, 2000, p. 125). Evaluative beliefs contain strong affective and evaluative components - whether concepts are "good" or "bad" or lead to "good" or "bad" (Abelson, 1979). The evaluative baseline of "good" or "bad" can be influenced by referents (e.g., relevant people, groups, and organizations). For instance, some ISD professionals are firm believers of executing projects that are on-time and on-budget on-time and on-budget, while others seek to add value to society. As reported by Kumar and Bjorn-Andersen (1990), ISD professionals' evaluative beliefs lead to different developmental processes and outcomes. They proposed the following key evaluative beliefs: *technical, economic*, and *socio-political beliefs*. Technical beliefs explain designers' focus on documentation, standards, procedures, and responses to changes; economic beliefs drive designers' attention to planning, control, budget, schedule, and human resources. Socio-political beliefs, on the other hand, are more related to behaviors such as participation, power, and governance.

In this dissertation, I associate knowledge and beliefs in a mental model context. According to "Organization Corollary" of Personal Construct Theory (Kelly, 1955), multiple concepts are organized hierarchically in mental models where beliefs are at the superordinate level. Beliefs can be observed through implied meaning of the concepts or the structure. For instance, the underlying belief of "Team and culture building" is people. "Team and culture building" and other concepts that share similar meanings, such as “Empowerment” and “Motivating and managing performance”, are likely to cluster together. The associations manifest one’s belief system, including the degree of importance (i.e., if you possess more concepts with similar underling meanings, you have a strong preference toward the belief). Hereafter, when I refer to beliefs, I am referring to beliefs in the mental model context. That is, I study evaluative belief systems which are represented by the structure of mental models.

It is worth noting that some research refers beliefs to cause-effect beliefs, which is about what effective tactics should be employed to achieve certain goals (Chattopadhyay et al., 1999). For instance, Irani et al. (2002) suggested a causal mapping approach to model the causal relationships between different concepts of IS project investment considerations (across strategic, tactical, operational, and investment domain). They argued that the approach is complementary to traditional quantitative evaluation models. The cause-effect linkage among concepts of investment considerations discussed in their paper should be considered task-specific knowledge (i.e., IS evaluation) rather than evaluative beliefs. Another example for the cause-effect beliefs is Passos et al.’s (2011)interview with Scrum Masters and Product Owners. They constructed the cause-effect beliefs between Scrum practices and project outcomes (e.g., the team-based story points technique[5] is related to accurate estimation and better understanding of requirements upfront). The knowledge should also be considered task-specific knowledge.

---

[5] Story requirements are ranked by points rather than hours. A team will begin with writing story requirements on an index card. Then, team members will discuss each story requirement and decide the order of each. Stories will be divided into smaller stories during discussion in order to be sized. Once the order of stories is decided, points will be assigned to each story. The efforts for story points are estimated based on the story with same points in previous iteration(s) or projects conducted by the same team.

2.4 Summary and Reflection

Table 1 summarizes research on ISD mental models[6] and the four themes mentioned in Section 1.2 (identification of the content of mental models, accuracy of mental models, similarity of mental models, and shifts in mental models). Not surprisingly, ISD research focuses on task-specific knowledge, with special attention to technology (e.g., programming, data structure algorithms, and database) and performance of ISD processes (e.g., technical communication and requirement construction). Researchers are particularly interested in the mental model of objective-oriented programming because it represents concepts adapted from traditional procedural languages and requires a major mental shift. They examine the accuracy of mental models for ISD experts and explore ways to alter the mental models of traditional procedural languages. (D. J. Armstrong & Hardgrave, 2007; Shaft et al., 2008). Research on requirement construction is also prevalent, and the mental model perspective is useful for eliciting accurate requirements. The visual representation of mental models helps convey each team member's understanding of the requirements, which in turn allows the team to clarify and refine them.

Table 1 Summary of Mental Models Research Related to ISD

	Task-specific knowledge	**Understanding of project members**
Technology knowledge	■ Conceptual modeling: (1) Siau & Tan (2005a) [Accuracy] (2) Siau & Tan (2008) [Accuracy] ■ Objective-oriented programming: (1) Sheetz and Tegarden(2001) [Accuracy] (2) Nelson et al. (2002) [Accuracy] (3) Armstrong & Narayanan (2005) [Definition] (4) Armstrong and Hardgrave (2007) [Accuracy and Shift] (5) Shaft et al. (2008) [Accuracy & Shift] ■ Programming: (1) Lau & Yuen (2010)[Accuracy] ■ Database: Siau & Tan (2006) [Accuracy]	N/A

[6] I did not list task-related knowledge because there is a dearth of research on them, if not none. Mental models about evaluative beliefs are not associated with knowledge and therefore are discussed separately later.

Domain knowledge	N/A	N/A
IS application knowledge	■ B2C website design: Tan et al. (2009)[Accuracy]	N/A
Organization knowledge	N/A	N/A
ISD process knowledge (Performance of ISD)	■ Technical communication: Siau & Tan (2005b) [Accuracy] ■ Requirement construction: (1) Montazemi and Conrath (1986) [Accuracy] (2) Byrd et al. (1992) [Accuracy] (3) Zmud et al. (1993)[Accuracy] (4) Hassenzahl and Wessler (2000) [Accuracy] (5) Davidson (2002) (Shift) (6) Browne & Ramesh (2002) [Accuracy] (7) Kudikyala & Vaughn (2005) [Accuracy & Similarity] (8) Niu & Easterbrook (2006) [Similarity] (9) Chakraborty et al. (2010) [Accuracy & Similarity]	N/A
ISD process knowledge (Management of ISD[7])	■ ISD methodology: Lee & Truex (2000) [Accuracy and Shift] ■ Critical factors of the management of ISD (1) Moynihan (1996) [Definition] (2) Moynihan (1997) [Definition] (3) Pankratz and Loebbecke (2011) [Definition]	N/A
	ISD Project Knowledge: this dissertation [Definition and Similarity]	
ISD knowledge in general	N/A	■ Expertise (1) Levesque et al. (2001) [Similarity] (2) He et al. (2007) [Similarity] (3) Espinosa et al. (2007)[Similarity] (4) Hsu et al. (2011) [Similarity] ■ Team interaction (1) Hsu et al. (2011) [Similarity] ■ Presence awareness (1) Espinosa et al. (2007) [Similarity]

[7] I omitted research that examined ISD professionals' mental models pertaining to the attributes of project members, such as the study by Napier et al. (2009), where critical project management skills were elicited through interviews with IT project managers. Although an understanding of required skill sets contributes to hiring, training, and ultimately, project success, such knowledge does not fit the definition of ISD process knowledge.

Note: Text in the bracket refers to the theme (s) of mental model research. Accuracy: accuracy of mental models; Similarity: similarity of mental models; Definition: identification of the content of mental models; and Shift: shift in mental models.

In addressing gaps within existing ISDP mental model data, I agree with Iivari et al.'s (2004) suggestion that IS application knowledge and ISD process knowledge should represent two distinct bodies of knowledge for the IS discipline, and that both deserve special attention. By accumulating distinctive IS knowledge, we as researchers can offer solutions for IS practitioners (Benbasat & Zmud, 1999; Keen, 1980). As discussed in Section 1.1, practitioners face challenges in ISDP management. For this reason, this study focuses on the management aspect of ISD process knowledge, specifically the knowledge within the ISDP context. Hereafter, we refer to this specific type of ISD process knowledge as *ISDP knowledge*. It should be noted that ISD projects can never be successful without technology knowledge, organization knowledge, application domain knowledge, and IS application knowledge. Figure 4 illustrates how ISDP knowledge helps a project team integrate the other four types of knowledge residing in different stakeholders' minds.

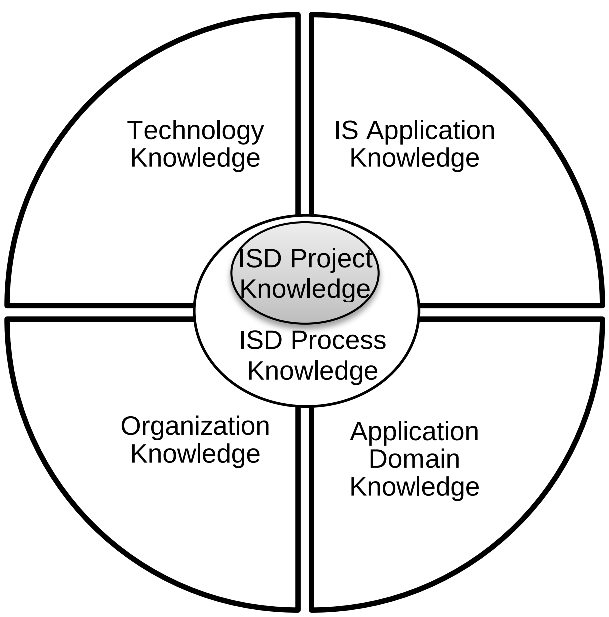


Figure 4 The Role of ISD Project Knowledge

The existing literature has introduced cumulative ISD project knowledge, such as technological aid (e.g., integrated case tools, Maxwell & Forselius, 2000), process management (e.g., CMMI, de Oliveira, Valle, & Mahler, 2010), behavioral factors (e.g., user participation and involvement, McKeen, Guimaraes, & Wetherbe, 1994), risk factors (Barki, Rivard, & Talbot, 2001), control factors (Kirsch, 1996), organizational factors (e.g., IT capability and resource adequacy, Butler & Murphy, 2008), social issues (Guinan, Cooprider, & Faraj, 1998), just to name a few. The question that remains unanswered is how ISD professionals know and believe in ISD project knowledge. We still have little understanding of mental models on ISD project knowledge. In Table 1, I highlight where this dissertation fits in the gap in the existing literature (see bold and underlined text). Study 1 defines the content of project members' ISDP mental models while study 2 explores the work relationship issues around the similarity of ISDP mental models between project managers and developers. Moreover, study 2 should fill another gap in IS research on mental models by advancing our knowledge to an understanding of other project members' mental models. The next section provides more elaboration on the gap and potential bridge for it.

An Understanding of Project Members' Mental Models. Table 1 also indicates few studies on an understanding of project members. This stream of research comes from the team mental model (TMM) construct. The application of the TMM construct in ISD is still in its infancy and has shown slow progress because the barriers to such work require substantial efforts to overcome. In order to advance this type of research, I suggest reconsidering the theoretical meanings of understanding and measurement of understanding.

From the theoretical perspective, prior research on understanding has focused on ISD knowledge in general without classifying ISD knowledge into different dimensions. The classification deepens our knowledge about different types of understanding and would lead to different consequences. For instance, an understanding of other project members' technology knowledge may entail different collaboration behaviors in comparison with an understanding of ISDP mental models. The former may assist in the identification of expertise and the assignment of tasks accordingly while the latter may help with coordination of project processes. Without a clear delineation of the understanding construct, it would be challenging to examine various impacts of understanding. In study 2, I will show different dimensions of understanding needed in the ISD project context.

A methodological concern for the current research on understanding lies in its predominant use of perceptual measure either by the direct consensus model or by the reference-shift consensus model (Chan, 1998). In the direct consensus model, individuals are asked to respond to their understanding of team members (e.g., I am well aware of other team members' skills and abilities) and uses within-group agreement indices to confirm the consistency. The reference-shift consensus model shifts the reference from the individual to the group, such as "team members knew what task-related skills and knowledge they each possess". The reference-shift consensus model aims to capture shared mutual understanding between project members without taking individual differences into consideration. Put differently, researchers will not know who have higher/lower understanding. However, both approaches lack confirmation about whether the understanding is accurate (i.e., Team members A thinks that he/she has an accurate understanding of team member B. However, it is possible A's understanding is inaccurate.) and do not contain information about to what extent the understanding is accurate. The mental model approach this dissertation adopts can resolve the issues. By understanding each team members' mental model, there is a potential to match one's understanding of another. A sample item can be "I know that team member X hold knowledge about defect prevention". Then, the response can be compared to team member X's mental model to assess the accuracy. Multiple items can be generated to assess the degree of understanding of the specific team members.

Evaluative beliefs. With the exception of a few studies (e.g., Johri & Nair, 2011; Kumar & Bjorn-Andersen, 1990), prior research has rarely examined evaluative beliefs for ISD per se (i.e., ISD values). Rather, national and organizational cultures are prevalent in ISD research and serve as proxy variables for a developer's major value system. For instance, high power distance culture hinders user participation and enhances management's power (Shore & Venkatachalam, 1995); high uncertainty avoidance culture requires more control over system access and development processes (Shore & Venkatachalam, 1995) and induces higher perception of project risks (M. Keil, Tan, et al., 2000). It is understandable for a widespread focus on national culture as there is ubiquitous phenomenon of globally distributed ISD project teams with diverse national cultures.

In a similar vein, substantial research has contributed to organizational culture values. For instance, research examined the compatibility between different organizational values and ISD methodologies (e.g., Iivari & Huisman, 2007; Iivari & Iivari, 2011; Ngwenyama & Nielsen, 2003; Tolfo & Wazlawick, 2008). It is believed that organizational values provide norms about the way the organizations do business. Namely, they help members judge what is appropriate when facing problems of external adaptation and internal integration. Through the attraction-selection-attrition process (B. Schneider, 1987), socialization (Salancik & Pfeffer, 1978), leaders' influences (Schein, 2004), and critical events (Cameron & Quinn, 2006), individuals values and organizational values are likely to be synchronized. For instance, people who work in a democratic organizational culture are more likely to embrace Agile approaches (Siakas & Siakas, 2007).

In both streams of research, it is assumed that national and organizational cultures can play a dominant role in ISD professionals' value systems. Nonetheless, although cultures affect and shape one's value systems related to ISD (Kankanhalli, Tan, Wei, & Holmes, 2004), underlying national and organizational culture goals (e.g., collectivism and power distance) differ from those underlying ISD (e.g., concerns about design and product). . Therefore, in order to understand the desirable ISDP outcomes that project members strive to attain, I suggest examining project members' ISDP value systems via a mental model perspective. Once the ISDP value system is identified, it can provide insight into why project members follow or resist certain procedures and practices, and also why conflict may arise among project members.

Figure 5 shows three areas in mental models in which we have little knowledge (see boxes and circles in the plaid pattern). In order to gain a better understanding about how project members' mental models work in ISD projects and in turn affect work relationships, the following two studies are devoted to filling the gaps in the existing literature and contributing to cumulative knowledge with regard to ISD project management.

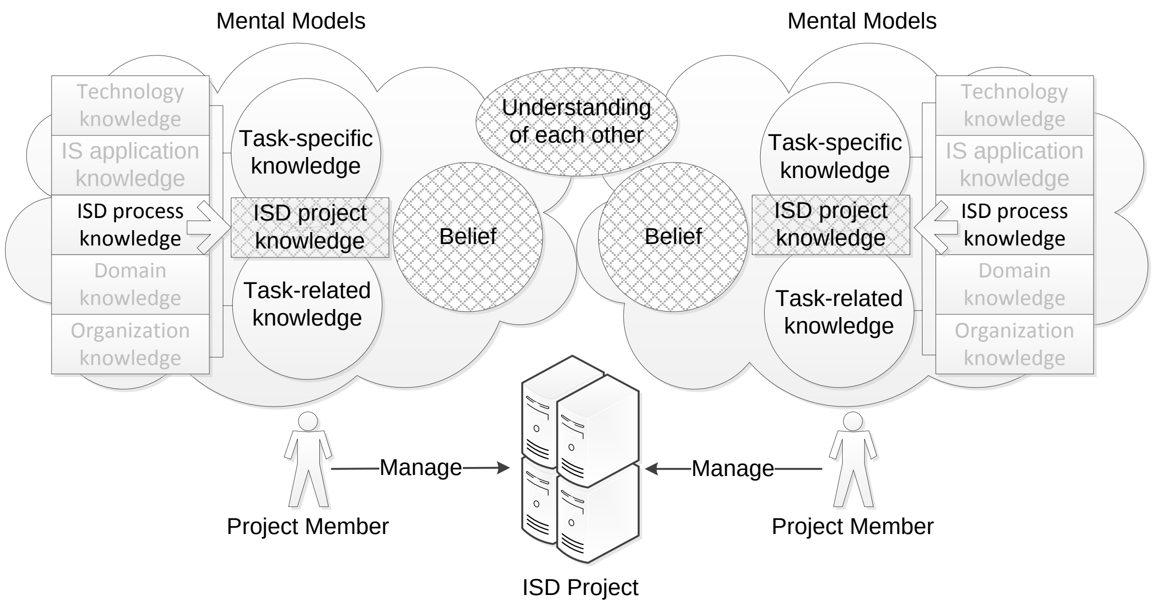


Figure 5 Research Gaps and Foci of this Dissertation

Chapter 3

Study 1: An Exploration of ISDP Mental Models

ISD professionals are confronted with a variety of challenges in ISD projects ranging from planning, analysis, design, to construction. In such an uncertain and complex work environment, ISD professionals have to tackle ISD issues based on the repertoire of knowledge at their disposal and apply it to solve problems. As stated in the Chapter Two, the knowledge I focus on is ISD project knowledge - how tasks and people in ISD projects should be managed. The repertoire of knowledge as well as its underlying beliefs is the core of the discussion in this chapter.

To define ISDP mental models, study 1 draws on the tools and techniques developed over several decades in cognitive psychology literature. The basic element of the content in a mental model is a concept - a meaningful unit in memory- and it is composed of a class of objects/instances that share similar characteristics (Smith, 1990). ISDP concepts can be distilled from a variety of technical and managerial practices and solutions (e.g., focusing on user participation and involvement, maintaining simplicity of design) and then be instilled into one's ISDP mental model. The relationships among these concepts specify the structure of the project member's ISDP mental model. Interrelationships between concepts are gradually shaped from learning, training, interaction with other stakeholders, and project experience. Project members therefore develop implicit theories for understanding and managing ISDPs. Put differently, the ISDP mental model with salient concepts and their interrelationships act as a filter for project members to making sense of all activities during the ISD processes. It helps project members to explain the how's and why's of events. This affects ensuing expectations and actions. For example, a project member could have a strong belief (cause-effect) that strict control over developers and users (a concept, which possibly includes a set of instances such as daily job journals and detailed meeting minutes) is essential to risk management (another concept). The whole statement presents an implicit theory; project members who hold such concepts and an implicit theory tend to prefer tight supervision. These concepts may be rooted in his or her mind. Deeply-rooted knowledge may evolve slowly especially when shifting to new concepts with fundamental differences (D. J. Armstrong & Hardgrave, 2007). The static characteristic promises the potential to analyze the content and structure of ISDP mental models. I reiterate the research question this study attempts to address below:

> *What is a project member's ISDP mental model? What are the core ISDP concepts that comprise a project member's knowledge and belief structures about the management of ISD projects and how are these concepts interrelated?*

Before research methodology, I will set up the context by elaborating on the relationship among ISDP mental models, ISD methodology, and methodology-in-action, as well as elucidating the two major components of mental models (content and structure). I begin with a brief introduction of what ISD methodology and methodology-in-action are and the relationship between them. Then, I discuss why people react to methodology-in-action differently using the mental model perspective. After that, the content of the mental model is explicated. I explain why methodology is a potential source of mental models. A review of the literature ends with the discussion about the structure of mental model. Figure 6 shows different components of ISD and the sequences I will introduce in the following sections.

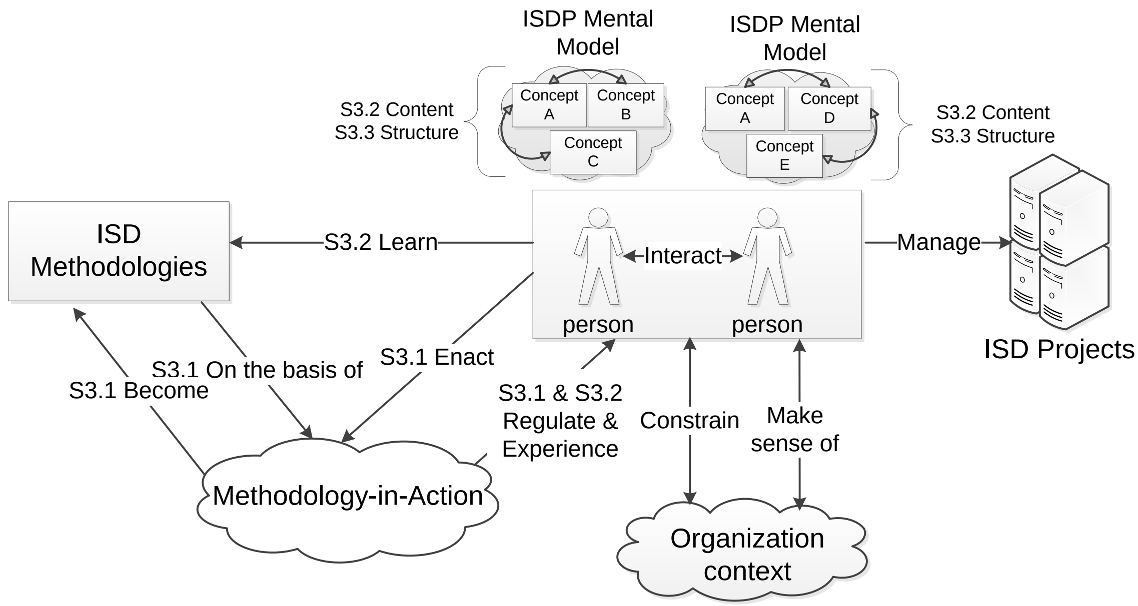


Figure 6 Mental Models and Methodologies

3.1 ISD Methodologies, Methodology-in-action, and People

ISD Methodology. An ISD methodology contains "a collection of procedures, techniques, tools, and documentation aids which will help the system developers in their efforts to implement a new information system" (Avison & Fitzgerald, 2007, p. 24). Besides, each methodology has its underlying philosophies, beliefs, and assumptions (Iivari, Hirschheim, & Klein, 1998), although they are not often explicitly stated. Object-oriented methodologies, for example, contain four phases of life cycle: inception, elaboration, construction, and transition; it suggests object-oriented analysis and design techniques; and it is supported by UML tools, such as IBM Rational Rose (Booch et al., 2007). Object-oriented methodologies make an assumption that requirements are unstable due to changing businesses. The principal ideas of abstraction, inheritance, encapsulation, and polymorphism in object-oriented analysis and design indicate two fundamental management concepts underpinning object-oriented methodologies:

"iterative and incremental development" and "reuse". The fundamental concepts thus act as the guiding principles, governing the technical development and management processes.

Methodology-in-action. Few ISD practitioners are methodology evangelists who would follow methodologies faithfully. Only 6% of them strictly practice the principles and rules that are set by methodologies (B. Fitzgerald, 2000). Instead, they formally or informally enact their own methodology for their projects on the basis of ISD methodologies[8]. Tailoring methodologies to a specific project or organizational context is common (59% in Fitzgerald et al.'s (1999) survey), which is called "methodology-in-action". From time to time, methodology-in-action becomes ISD methodology because of its usefulness and popularity; for instance, eXtreme Programming (XP) (Beck, 2000) is initially a methodology-in-action used in the Chrysler Comprehensive Compensation System (C3) payroll project.

Two distinct fundamentals are behind methodology-in-action. One is an adaptation of an existing methodology by combining other methodologies that share similar principles; the other is a mix of methodologies as diverse as possible if needed. The former approach tweaks the existing methodology and includes practices from methodologies that hold similar goals. Take XP and Scrum for example, the tweak leads to either various versions of methodologies, such as XP-lite (Aveling, 2004) and SCRUMBut (Krishna & Basu, 2011), or a combinitaion of XP and Scrum (B. Fitzgerald, Hartnett, & Conboy, 2006). The assumption of the later adaptation approach is that a single methodology or a combination of similar methodologies may be too narrow to address multiple aspects of ISD issues (Avison, Wood-Harper, Vidgen, & Wood, 1998). The idea can be seen in a synthesis of plan-driven and agile methodologies for achieving ambidexterity (Boehm & Turner, 2004). These two methodologies espouse different principles and values. Integration of these two helps organizations obtain rapid business value and be responsive to

[8] ISD methodologies are referred to methodologies that are formally documented and have been widely recognized and discussed in ISD communities, such as Rational Unified Process by IBM or Capability Maturity Model by the Software Engineering Institute.

change while at the same time maintaining predictability and stability; for instance, organizations reconcile principles from both methodologies to maintain control in distributed software, such as formalized communication and specifications, while ensuring the existence of informal collaboration to obtain agility during different development contexts (Ramesh, Mohan, & Cao, 2012).

Although methodology-in-action is intended to meet situational needs, it is created by a small group of members in organizations and often it is not well received by other project members. Alistair Cockburn, an experienced ethno-methodologist who has studied the ethnology of ISD methodologies in fields for over two decades, has offered an observation.

> *"Whenever I go in with a prescriptive recommendation, teach people to work this way, basically they don't. They work, however they work, inside their own heads, and a few people will pick up a few bits of a few of the techniques."*- interviewed by Highsmith (2002)

Such resistance can come from two factors related to mental models: insufficient knowledge and unfulfilled benefits of methodologies. In other words, it depends upon whether methodologies are internalized in project members' minds and whether espoused values of methodologies are aligned with project members' evaluative beliefs. Seemingly, the root of resistance is people – more precisely, their mental processing and mental representation. The two factors behind resistance will be discussed in details below.

Methodology-in-action and Mental Models. Some methodologies, such as object-oriented methodologies, underline in-depth technology knowledge, while others, such as Soft Systems Methodology (Checkland & Scholes, 1990), stress the alignment between ISD projects and organizational strategy where organization knowledge is essential. Still others demand people skills and highlight the importance of user participation, such as Effective Technical and Human Implementation of Computer-based Systems (ETHICS) (Mumford, 1995). Different methodologies require different knowledge. ISD practitioners are in favor of methodologies by which their knowledge and skills meet the demand because the adoption of those methodologies requires less cognitive efforts and has been proven to be effective in their prior projects (Kristof, 1996; Rokeach, 1960). While learning is a possible way to reduce resistance and improve adoption, the literature has suggested people tend to face challenges in learning different development paradigms, such as a shift from structured methodologies to object-oriented methodologies (Shaft et al., 2008).

Even when ISD practitioners have a solid understanding of methodologies, they show resistance to methodologies because they are not compatible with their values and thus fail to experience the benefits of successful adoption (Avison & Fitzgerald, 2003; Hardgrave, Davis, & Riemenschneider, 2003; Riemenschneider, Hardgrave, & Davis, 2002). Iivari and Huisman (2007) showed that IT managers who worked in achievement-oriented organizational culture did not believe in the benefits of traditional methodologies on productivity, efficiency and goal achievement, thereby affecting their acceptance and use of methodologies. Although Iivari and Huisman did not examine individual values per se, it implies that those IT managers possess specific values toward ISD (very likely, values on productivity, efficiency and goal achievement), which traditional methodologies implemented in their organizations cannot fulfill. Passos et al. (2011)observed different beliefs about Scrum software practices where Scrum Master insisted on no requirement change during the Sprint session because he/she believed productivity is the key while the Product Owner suggested that each Sprint session should reserve time for customers' issues on account of the importance of customer satisfaction[9]. Perhpas, ISDP mental models in terms of beliefs

[9] This case is related to the importance of understanding of other project members' mental models I alluded to it in Section 2.4. The Scrum Master could have convinced the Product Owner if he/she understood the Product Owner's belief. The fundamental idea behind the adherence to planned activities and enforcement of sprint deadlines aims to continuously move project forward. While responding to customers' issues is a good goal, it is suggested that disciplined management of issues rather than random reaction to requests would be beneficial for schedule overall. Customers' issues should be put into backlogs and then prioritized into the following Sprint sessions. Having said that, customers' issues still can be resolved on the date project team promised, without jeopardizing important project schedule. That means sustained customer satisfaction.

played an essential role in the above scenarios when project members responded to different practices and methodologies.

The above discussion demonstrated the intertwined relationships among methodology, methodology-in-action, and people. It suggested that organizations struggle to realize benefits promised by methodologies when methodologies are incompatible with project members' knowledge and beliefs. Even when methodologies are followed and formal (e.g., procedures, rules, and goal setting) and informal control (e.g., social norms) is delivered, one should be cautious whether desirable project outcomes can be obtained. Given the complexity of ISD projects, project members continue facing new challenges. Methodologies alone, even for the ones with high agility and flexibility, may not prepare them to cope with these challenges. Ultimately, the success of a project lies in project members who are able to adapt methodologies to the particular needs and demands of tasks. The ability to adapt primarily depends upon project members' mental models. Project members have to internalize best practices so that they can act upon these practices. Having said that, I have no doubt methodologies are valuable, but I suggest that their roles should be more similar to a template for the management of ISD projects: a solid bedrock where project members can build upon their unique, but agreed-upon, way in meeting work challenges. In the next two sections, I will explicate project members' mental models that grant individuals with mental dexterity for ISD projects. Section 3.2 touches upon the content of mental models where concepts are building blocks. I explain why methodologies should be a major source of mental models. Section 3.3 suggests the combinations of common concepts manifest in the structure of mental models. I propose that evaluative beliefs shape the pattern across diverse combinations

3.2 Content of ISDP Mental Models

ISD project knowledge is crystallized from past experience (e.g., training, working with people, and project implementation) through a series of inferences, including generalization, abstraction, explanation, and similization, according to the inferential theory of learning (Michalski, 1993). New instances during ISD processes are subsequently included in the concept if they fit into the attributes of an ISDP concept or a set of ISDP concepts. It is important to note that concepts are the building blocks of the content of mental models (Kunda, 1999).

Concepts can be formed at a different level of abstraction according to the hierarchical principle of cognitive psychology. This study emphasizes fundamental concepts – knowledge about the management of ISD projects at the higher abstract level. Different from work practices -"real" ways of doing things (Iivari & Iivari, 2011), fundamental concepts are more powerful to understand decision-making and behaviors than work practices (or concepts at the lower hierarchy level of abstraction) because they are more static and representative of the essential spirit of work practices. For example, one of the fundamental concepts of Agile methodologies is a focus on people and collaboration during development. There are a variety of work practices supporting this concept, such as XP's pair programing or Scrum's working in a common project room. People may develop their understanding of concepts as they continuously apply these work practices (Kensing & Munk-Madsen, 1993). Although specific practices that can realize the benefits of fundamental concepts may be useful practical knowledge for ISD professionals, I suggest that a focus on the fundamentals should be of theoretical and practical importance. Work practices connote multiple meanings; for instance, pair programming belongs to both the concept of "close collaboration and communication between project members during development" and the concept of "defect prevention". As a result, it is challenging to accurately assess one's knowledge and beliefs.

Unless we understand how individuals see the world of development (i.e., fundamental concepts and the relationships), organizations would find it difficult to persuade them into learning and applying practices. Besides, since there are a finite number of fundamental concepts within mental models (Kelly, 1955), it provides a reasonable foundation to develop the ISDP mental model construct with a balance between comprehensiveness and parsimony, thereby leading to a construct with higher clarity.

ISD methodologies, both software engineering oriented and project management oriented[10], can serve as a starting point to unearth fundamental ISDP concepts because an ISD methodology is made up of accumulated work practices based on IS experts' experiences (Avison & Fitzgerald, 2007). Both academics and practitioners have kept searching for the methodological holy grail because they believe that the "right ISD methodology" will set right directions and rules of action, resulting in productivity and quality gains. Even though ISD professionals do not follow IS methodologies doctrinally (B. Fitzgerald, 1998), their ways of managing ISD should not deviate considerably from ISD methodologies because training, communication, socialization, and institutionalization have set the boundaries of ISD professionals mindsets.

There have been continued efforts to organize knowledge relevant to the management of ISD via sorting out a myriad of methodologies (Avison & Fitzgerald, 2007; Iivari et al., 1998, 2000). Applying the idea of object-oriented decomposition, Iivari et al. (1998, 2000) decompose methodologies into four inherited domain objects. The most abstract object of methodologies is paradigms, which hold different epistemologies and ontologies for understanding methodologies. Object at the second level is ISD approaches that group common goals, guiding principles, fundamental concepts, and principles of ISD. At the lower level, the ISD methods capture detailed ISD processes. At the lowest level, ISD

[10] The software engineering oriented methodologies focus on the software construction process such as waterfall, spiral, and object-oriented methodologies while the project management oriented methodologies tend to facilitate the construction process in order to complete on time, on budget, within scope, and without defects, such as project management body of knowledge (PMBOK), and PRojects IN Controlled Environments 2 (PRINCE 2) (Chemuturi & Cagley Jr, 2010). Nowadays the distinction between these two orientations is generally blurred. Either orientation blends the ideals from one another. For instance, Agile methodologies, such as dynamic systems development method (DSDM) and Scrum, pay attention to cost, quality, and schedule using agile-oriented work practices such as daily stand-up meeting and time box deadlines. The project management component is weighed in these methodologies. Since both software engineering and project management approaches shape one's mental model, I will include both regardless of their engineering or management focus, only if the goal of these methodologies is to help manage and implement IS.

techniques/tools used to support specific methodologies are recognized. Similarly, Avison and Fitzgerald (2007) develop a framework that contains seven elements of methodologies. Consistent with Iivari et al. (1998), the framework is mainly comprised of philosophies, principles, and techniques/tools. These frameworks provide insights into the identification of fundamental ISD concepts. Therefore, when reviewing ISD methodologies, I focus on those elements suggested by prior research and summarize fundamental concepts from them (See Section 3.4.1).

However, we should be aware that accumulated knowledge in ISD methodologies would not directly contribute to our understanding of project members' ISDP mental models because the above taxonomy of ISD project knowledge is too "objective" (or academic) to resemble the arrangement of knowledge in practitioners' minds. Practitioners are less likely to process knowledge into such a systematic and codified format through rote memorization of isolated facts. Rather, they reason different principles and practices of ISD and form concepts from reasoning processes. Thus, fundamental concepts derived from ISD methodologies only act as a baseline. To reveal what constitutes project members' mental models, I glean concepts from the sample of ISD professionals using various mental model elicitation techniques (See Section 3.4). Once concepts are derived from ISD professionals, I compare the concepts that ISD professionals hold and the pool of fundamental concepts coded from ISD methodologies. This approach validates comprehensiveness and avoids missing some fundamental concepts.

3.3 Structure of ISDP Mental Models

Structure refers to the systematic arrangement of concepts within a mental model. The organization of concepts helps determine how an ISD professional absorbs new concepts, makes inferences, and acts upon concepts in a particular work context. The structure reveals the concept pathway(s) that an ISD professional will tend to follow. According to the learning literature, the structural connections among concepts are driven by evaluative beliefs (Barsalou, 1991; Rothkopf & Billington, 1979); People are receptive to concepts that help them achieve their goals; accordingly, those concepts naturally tend to cluster together. In other words, evaluative beliefs determine how knowledge is organized and subsequently regulates one's behaviors. Therefore, we expect evaluative beliefs to be evident from the arrangement of ISDP concepts.

Despite the long-standing research on evaluative beliefs, or values (hereafter we use the terms interchangeably, as *value* is a prevalent term in psychology and other disciplines), few studies have investigated them in the ISDP context. Values have been the focus of research in various social science disciplines, including sociology, psychology, politics, anthropology, and management. Moreover, scholars have proposed several types of value systems, such as Universal Value (Schwartz, 1992), Competing Organizational Values (R. E. Quinn & Rohrbaugh, 1981), Minnesota Importance Questionnaire (Lofquist & Dawis, 1978), and Work Value Inventory (Super, 1970). Prior research in other fields, however, cannot lend direct support as to why ISD professionals learn and accept certain types of ISDP knowledge. To the author's knowledge, Kumar and Bjorn-Andersen (1990) are the only authors that discuss ISD values[11]; their three evaluative beliefs–economic, technical, and socio-political–shed light on the ISDP mental model structure.

Economic belief, concerned with the management of resources, is closely related to project constraints (cost, schedule, and scope). Economic beliefs drive people to adopt work practices that focus on planning, evaluating, and regulating ISD activities (Andersen et al., 1990). For instance, planning can include an estimation of software development efforts through either informal (e.g., expert judgment) or

---

[11] Some researchers discuss ISD values, but do so within a constrained context. For instance, Johri and Nair (2011) studied an e-government project in India and showed the importance of ISD values on transparency for the reduction of corruption/and showed the importance of ISD values of transparency to reduce corruption. Additionally, given its status as a limited resource, values for money/the ISD value of money are also salient. This type of research is difficult to generalize and is not included in the discussion.

formal (e.g., work breakdown structure, function point analysis, COCOMO, XP planning games) work practices. This estimation can then be used to define the schedule and budget, and to monitor progress. ISD professionals whose responsibility is project management, such as project managers and IT managers, may hold economic beliefs.

Technical belief emphasizes the management of technical components. The desirable outcome is a reliable, adaptable, and high-quality software product. People who hold the technical belief advocate for technical activities, such as analyzing technical options and organizational contexts, formulating a blueprint for a desired change within an organization, and implementing appropriate systems. In short, analysis, design, and realization are three primary activities that ensure functioning software (Andersen et al., 1990). Constant refactoring to improve code quality illustrates one technical belief work practice. ISD professionals who have experience in system construction, such as programmers, system analysts, and business analysts, are more likely to hold a technical belief.

Socio-political belief is deeply involved with the interests of stakeholders, including developers, users, and champions and sponsors. Regarding developers, this belief is concerned with their well-being, interpersonal relationships, and productivity within an ISD environment, and focuses on work practices, such as performance measurement, work arrangements, and training (Agarwal & Ferratt, 2002). Empirical research has revealed that work practices related to this belief include training (Jiang, Sobol, & Klein, 2000; R. R. Nelson, 1991), team development, and leadership (Jiang et al., 2000). With regards to users, this belief considers a customer's problems and issues, and the desirable outcome is customer satisfaction. When considering champions and sponsors, this belief strives to enhance champions' and sponsors' understanding of the project and gain their support to ensure a smooth project trajectory

These three beliefs provide a lens to understand the ISDP mental model structure. We must emphasize that we do not propose individuals hold only one belief. On the contrary, it is likely that experienced ISD professionals appreciate multiple beliefs which results in a more balanced mental model. The present study investigates whether ISD practitioners possess these three beliefs and if so, the linkage between these beliefs and ISDP concepts.

3.4 Research Methodology

Scholars and practitioners have long sought to precisely measure and represent complex mental models (Langan-Fox, Code, & Langfield-Smith, 2000; Mohammed et al., 2010; Smith-Jentsch, 2009), a challenging goal due to the multifaceted nature of mental models. As explained above, content and structure comprise two critical aspects of mental models in any given problem domain. Only when the content is properly defined and the structure is adequately captured, can one achieve a highly valid knowledge structure construct and meaningfully examine its relationships with other constructs within its nomological network (DeChurch & Mesmer-Magnus, 2010; Suddaby, 2010). The content and structure of mental models are best studied by employing elicitation and representation techniques. Elicitation refers to the approach used to establish the elements or content of a given mental model, whereas representation refers to the approach used to understand the structure of the data or, put more simply, how these elements relate to one another within an individual's mind (Mohammed et al., 2000).

I performed two studies (Study 1a and 1b) to define the content and structure of project members' mental models concerning ISDPs. In the first study, we elicited the content through interviews with ISD experts, where the Repertory Grid Technique (RGT) was used to generate concepts to complement those obtained from ISD methodologies. In the second study, I determined the structure through the use of the pairwise rating and multidimensional scaling techniques. More detailed discussion of each technique is presented below.

3.4.1 Study 1a: Eliciting ISDP Mental Models

Concepts within a content domain can be either supplied by researchers (i.e., nomothetic elicitation) or elicited directly from participants (i.e., ideographic elicitation) via cognitive interview and observation (see Appendix B for a summary of elicitation techniques). Ideographic elicitation gives researchers an opportunity to view divergence of mental models by encourage participants to provide as many concepts as possible. Nomothetic elicitation allows researchers to statistically compare different mental models because each participant works with identical concepts. This research combines ideographic elicitation with nomothetic elicitation. Ideographic elicitation is adopted because little is yet known what concepts exist in ISDP mental models. A concern about nomothetic elicitation is whether supplied concepts are comprehensive (Mohammed et al., 2000) and meaningful to respondents (Hodgkinson & Clarkson, 2005). To address this issue, I continued interviewing ISD experts until no new concept came from the ISD experts (i.e., the point of saturation). Then, I compared the pool of concepts from ISD experts with concepts derived from the literature review. For concepts not mentioned by participants, I asked participants whether these concepts are meaningful to them. The comparison ensures the comprehensiveness and meaningfulness of the list of concepts.

Once the content of ISDP mental models is defined, nomothetic elicitation is employed for assessing the structure of ISDP mental models. The following sections provide the detailed descriptions of the process of mental model elicitation.

3.4.1.1 Content Analysis

While researchers have attempted to capture fundamental concepts of ISD methodologies (Iivari et al., 1998), their purpose is to classify methodologies. Therefore the identified concepts are too abstract for practitioners to understand. For example, the concept of “Human activity systems” is considered a fundamental concept of Soft Systems Methodology, but practitioners may have difficulty understanding what precisely is meant by “Human activity systems”. To make such concepts more applicable to real-life situations, one solution is to ascribe them with more goal-driven descriptions. In order to understand complex “Human activity systems”, for instance, practitioners may incorporate concepts such as “Comprehensive requirement analysis (C19) that focus on the works and welfare of stakeholders and users (C8)”.

The discrepancy between theoretical and practical understandings of ISDP knowledge warrants a review of ISD methodologies and finding a means to adapt theoretical languages to actual ISDP circumstances.

I started with distilling concepts from ISD methodologies suggested by Iivari et al. (1998) and Iivari et al. (2000) and extended search and analysis to relevant ISD methodologies not included in Iivari et al.’s (1991; 1998, 2000; 2004) papers. The content analysis covers popular textbooks on system analysis and design (Kendall & Kendall, 2004; Whitten & Bentley, 2006), software engineering (Glass, 2003; Sommerville, 2004), IS project management (Highsmith, 2009; Murch, 2001; Olson, 2001), and practitioner’s books on ISD methodologies (Avison & Fitzgerald, 2007). I also reviewed seminal papers about the agile development approach (Abrahamsson, Oza, & Siponen, 2010; Abrahamsson, Warsta, Siponen, & Ronkainen, 2003; Highsmith, 2009; Williams, 2010) with associated methods such as XP (Beck, 2000), Scrum (Schwaber & Beedle, 2001), and Crystal (Cockburn, 2005). Furthermore, research on software development best practices was examined (Dutta, Lee, & Van Wassenhove, 1999). This compilation of concepts was not intended to be exhaustive. Rather, our objective was to establish a meaningful foundation of the essential concepts involved in ISD management.

I focused on fundamental concepts of current ISD practices and extracted 36 ISD concepts with over 80 subordinate concepts and practices (see Appendix A). The definition of each concept is described in Appendix H. The fundamental ISD concept is a relatively abstract concept, such as “Conscious efforts to make project size, cost, and schedule estimation”, while the subordinate concept and practice is a more concrete concept, such as function point analysis and work breakdown structure. Concrete concepts facilitate an understanding of abstract concepts (Kensing & Munk-Madsen, 1993; Medin, Lynch, & Solomon, 2000). The literature review establishes groundwork for understanding potential ISD concepts discussed in ISD communities. In the next section, I will introduce in-depth interviews with ISD experts in which I captured what concepts exist in ISD professionals' minds.

3.4.1.2 The Repertory Grid Technique

There are a variety of ways of eliciting concepts from subject matter experts through interviews, such as verbal protocol analysis, open interviews, inferential flow analysis, and visual card sorting, etc. I conducted interviews with the repertory grid technique (RGT) because of its high validity and test-retest reliability (Wright, 2008) and its appropriateness for addressing our first research question. Unlike other elicitation techniques, such as visual card sorting, the RGT deeply explores an individual's knowledge and belief structure via iteratively asking the participant to differentiate concepts in a given domain (Langan-Fox et al., 2000). Additionally, the stringent semi-structured interview process ensures that the participant focuses on a domain of interest, as compared to more open-ended interview techniques.

The RGT is based on personal construct theory (Kelly, 1955). Assuming that "Man is a scientist", personal construct theory proposes that people form hypotheses (if A, then B) and develop their own theories to explain daily events and anticipate actions. A theory is composed of constructs[12], more specifically "personal constructs". People always construe things and events by virtue of constructs. Constructs are always bipolar in nature with one preferred side (e.g., willingness to change vs. lack of flexibility in ISD) and are the most important elements in a grid. The RGT helps people uncover their personal constructs through a rigorous process and reduce research bias (A. Curtis, Wells, Higbee, &

[12] Constructs used in personal construct theory are synonymous with concepts used in other psychological literature. Two terms will be used interchangeably in this section.

Lowry, 2008; Langan-Fox et al., 2000). Despite the richness and accuracy of data from the RGT, the downside of the RGT includes time commitment needed from interviewees and high attrition (A. Curtis et al., 2008). To mitigate these issues, I recruited ISD experts who have rich knowledge and vested interests in the phenomenon investigated. Various interview techniques were adopted, such as a short break and encouragement tone when the researcher observed the interviewee was getting tired or frustrated. In the end, there was no dropout from the participants. The results of RGT are often a mix of diverse concepts, causing difficulties in interpretation. In order to summarize findings, I employed iterative coding procedures along with a coding schema developed by the content analysis. This approach ensured flexibility in coding concepts while at the same time maintaining the reliability of coding results.

Below I will describe the application of the RGT in eliciting ISD concepts from an IS expert's mental models. For more information about operational details of the RGT, see Stewart & Stewart (1981), Jankowicz (2004), and Fransella, Bell, and Bannister (2004); For discussion of the RGT in the IS field, see Tan and Hunter (2002), Curtis, Wells, Higbee, and Lowry (2008), and Edwards, McDonald, and Young (2009).

Participants. Participants for study 1 are ISD experts who have rich experience in ISD projects. The job roles include project managers, system/business architect, system/business analysts, programmers, testers, consultants, and other similar ISD project roles. I recruited participants via existing personal relationships, professional forums (e.g., Toronto XP users group[13], Agile Ottawa[14], LinkedIn), and the Canadian Information Processing Society (CIPS)[15]. I also sought participants through the snowball method by asking interviewees if they know and could lead to any other potential participants who are ISD experts. Their expertise was evaluated through a 7-point Likert scale, which asked them to

[13] http://tech.groups.yahoo.com/group/xptoronto/message/2302
[14] http://tech.groups.yahoo.com/group/agile-ottawa/message/208
[15] http://www.cips.ca/Volunteers-Needed-Managing-Information-Systems-Development-Projects-Exposing-the-Mosaic-Mind-of-IS-Professionals-June2012

indicate their confidence in their expertise in the following seven ISD domains: (1) project management, (2) business analysis, (3) system analysis, (4) business architecture, (5) system architecture, (6) programming, and (7) testing (1: Not at all confident, 7: Extremely confident). Additionally, the participants were asked to provide their years of experience in each domain.

Participants included in this study were those who had indicated "considerable confidence" (i.e., 6 out of 7 in a Likert scale) in at least one of seven ISD domains. A total of 19 ISD experts (16 men and 3 women) participated in the study. Their average IS/IT experience is 14 years and they have heterogeneous expertise. Table 2 presents their demographic information and expertise. The depth and breadth of their expertise permitted a more comprehensive capture of the ISD mental model content.

Table 2 Participant Demographics

Participant ID	Gender	Education	IS/IT experience (years)	Expertise
#1	Male	MBA	20	Project management (considerably confident, 10-year experience), Business analysis (considerably confident, 10-year experience), and) and System architecture (considerably confident, 15-year experience)
#2	Female	PhD in MIS	8	Project management (extremely confident, 9-year experience)
#3	Male	BEng in Mechanical Engineering	13	Programming (considerably confident, 10-year experience)
#4	Male	BSc in Software Engineering	4	Programming (considerably confident, 4-year experience)
#5	Male	MSc in Environmental Engineering	15	Project management (considerably confident, 20- year experience)
#6	Male	BSc in Actuarial Science and Computer Science	14	Programming (considerably confident, 14-year experience)
#7	Male	BSc in Biology	13	Project management (considerably confident, 13-year experience), Business analysis (considerably

				confident, 5-year experience)
#8	Female	MSc in Computer Engineering	12	Business architecture (extremely confident, 4-year experience) and Programming (considerably confident, 12-year experience)
#9	Male	MBA	12	Business analysis (considerably confident, 6-year experience), Business architecture (considerably confident, 3-year experience), and System architecture (considerably confident, 3-year experience)
#10	Male	MSc in Information Management	8	Programming (considerably confident, 8-year experience) and System architecture (considerably confident, 5-year experience)
#11	Male	MSc in MIS	7	Programming (extremely confident, 10-year experience) and Testing (extremely confident, 2-year experience)
#12	Male	BEng in Electronic and Instrument Engineering	11	Project Management (considerably confident, 8-year experience) and Programming (considerably confident, 7-year experience)
#13	Male	BBA in Accounting (minor in Finance)	43	Project management (extremely confident, 25-year experience), Business analysis (extremely confident, 15-year experience)
#14	Male	MSc in Computer Science	13	Programming (considerably confident, 10-year experience) and System architecture (considerably confident, 5-year experience)
#15	Male	BSc in Computer Science	4	Programming (extremely confident, 16-year experience)
#16	Male	BA in Data Processing	32	Project management, Business analysis, System analysis, Business architecture, System architecture, and Testing (considerably confident, each has 30-year experience)
#17	Male	BEng in Mechanical Engineering and Computer Science	8	Test (considerably confident, 6-year experience)
#18	Male	BEng in Control Engineering	15	Programming (extremely confident, 15-year experience) and Testing (extremely confident, 10-

				year experience)
#19	Female	BSc in Biology and Business Administration/ Information Systems diploma	12	Business analysis (extremely confident, 10-year experience) and Testing (extremely confident, 12-year experience)

Procedures. All participants were provided with an opportunity to review the letter of information and consent form before the interview. With the consent for participation, each participant went through three RGT tasks (see the RGT Tasks section). The majority of interviews (14 interviews) took place at a mutually agreed-upon location. However, when the face-to-face interview was not an option, a video interview via Skype was adopted. I used the screen sharing feature to provide visual support needed by the RGT. The interview lasted 81 minutes on average (*SD* = 12.88). After completing the interview, participants were thanked for their participation and received a $15 gift card.

To ensure the reliability and validity of the results, the repertory grid interview process was tested using a pilot. Five pilot tests of the repertory grid interviews were completed with faculty, doctoral students, and ISD professionals in the IT department of a Canadian university. Their feedback has been used to modify the research design.

RGT Tasks. RGT consists of three major tasks: element elicitation, concept elicitation, and linking elements with concepts.

(1) Element elicitation: The stage of selecting instances of a discussion topic refers to element elicitation. Elements can be people, objects, events, and activities (Stewart & Stewart, 1981), such as system analysts as an element for discovering ISD professionals' and users' perceptions about an excellent system analyst (Hunter, 1997), IS project managers as an element for learning IS project managers' required project management skill sets (Napier et al., 2009), and IS project as an element for

determining risk factors in IS project managers' mental models (Moynihan, 1996). The element in this study is an ISD project.

The participant was asked to identify six ISD projects he/she is working or worked on. It was explained that the projects should include successful as well as unsuccessful ones, so that a wider range of concepts can be elicited. The participant was reminded that the projects need not be very recent, nor do they need to have been completed, or be currently undertaken in the current organization. A total of 114 projects were provided by 19 interviewees. Table 3 summarizes the characteristics of these projects. The variability of these projects provides an opportunity to explore a variety of ISD project management concepts.

Table 3 ISD Project Characteristics

Project type	In-house new development, Packaged software development, Enhancement of existing software, Outsourcing software project, Customized modules, Research and development project, Proof-of-concept development.
Interviewee's role in a project	Business analyst, System analyst, Business architect, System architect, Project manager, Programmer, Senior business management, Senior IT management, Tester, Consultant.
The industry that a project is in	Consulting, Education, Finance and Insurance, Government , Healthcare, Manufacturing, Retail, Software, Telecom/network, Transportation, Media, Energy, Advertising, High technology.
Project team size	2 to 60 project members (53% of projects have 5 to 15 project members).
Project budget (CAD)	10,000 ~ 30 million (56% of projects have budget over 1 million).
Project duration	1 month to 10 years (63% of projects last between 3 months and 1 year).

(2) Concept Elicitation**:** Once elements are elicited, the elements can be used to elicit concepts. Among the various techniques to conduct this exercise, such as full context form (Reger, 1990) and group construct elicitation (Stewart & Stewart, 1981)), the most common technique is the triad elicitation method. The idea is to let the participant randomly select three elements from the element pool created at the element elicitation stage and to specify a way in which two are similar, but different from the third.

Accordingly, the participant provides a meaningful concept (s) with two bipolar phrases (e.g., the team climate is “respect others’ expertise” against “look down on others”). The participant continues to select triads until no new constructs are revealed. The average triad to reach saturation is between 7 and 10 (Reger, 1990). It is suggested that the triad process assists the participant in immersing himself/herself in the topic through continuous comparison. The triad process also avoids observer bias due to minimal intervention from the interviewer during the elicitation process.

In this study, I put six elements (projects) on index cards respectively and added two cards to the stack as anchors, with one labeled “ideal” and the other one labeled “suboptimal (unsuccessful)”. The two pseudo labels not only increase the contrast between elements but also help the researcher understand attitudes and beliefs about extreme cases in the ISD experts’ minds. The participant began with shuffling the deck of eight index cards and randomly selected three cards from the stack. Then the participant was asked: “Tell me in what important ways two of these three projects are the same, but different from the third, in terms of important managerial and technical practices for developing an information system in a project setting”. I encouraged the participant to think of the contrast (e.g., incremental development vs. linear development) rather than the negative (e.g., incremental development vs. non-incremental development) because the negative word provides little information about in what way two concepts are different (e.g., what does “non-incremental” mean?). The result of their answers was a pair of words or phrases used to describe both the similarity and the contrast (see Appendix D).

When the interviewee got stuck and could not offer any concept for a particular triad of projects, the researcher illustrated the procedure by suggesting a concept of his own (if this happened at the beginning due to unfamiliarity of the triad elicitation process) or asked the interviewee to pick another triad. I also assured that it is common and provided them with the encouragement about what he/she had done. When the triad approach did not work, I allowed interviewees to identify similarities and differences by sorting projects into piles from all eight projects (i.e., the full context form approach (Jankowicz, 2004)). In some cases, the interviewees would take a short break and had a brief chat with the interviewer about something else for a minute or two. When the participant provided a concept that is too vague or too specific, a "laddering" technique (Stewart and Stewart 1981) was employed to encourage the participant to elaborate on the elicited concept by incorporating a series of "how" and "why" questions. The "how" questions can be used when the construct is too general. The "how" questions seek to gain a deeper understanding of categorizations and produces a more specific construct (i.e., general to specific can be visualized as moving down the ladder, or laddering down). For instance, participant #16 suggested the concept "Clear communication between project stakeholders". I asked how communication helps project execution. In this case, clear communication referred to expectation management, where the project scope and goals are communicated at the start of the project, rather than at a later stage when irreconcilable differences may have surfaced. The "why" question aims to expose the core constructs in the participant's mind (i.e., specific to general means climbing up the ladder, or laddering up). The laddering up questions can make personal constructs more comprehensive and clarify their meanings. Two interviewees (participants #4 and #8), for instance, referred to a specific work practice known as pair programming that differentiates two projects from a third. By inquiring why pair programming is preferred, the researcher gained insight into higher level concepts, i.e., that pair programming promotes close collaboration and/or prevents poor design and defects.

Each interview ended at around the 4^{th} triad because participants could not offer any more new concepts. As compared to 7 to 10 triads in prior research (Reger, 1990), the fewer rounds of triad elicitation could be due to fatigue. However, I posit that this could be attributed to the participants' characteristics as well. The participants are experts in the domain. In each triad elicitation, they were able to think of many relevant concepts beyond the triad or even beyond the six projects. I did not stop participants from generating concepts that do not directly link to the triad since the purpose of the RGT is to elicit as many concepts as possible. To ensure that the participant did not miss any concept, in the end of the triad elicitation process, all six index cards were laid out before the participant along with all concepts that were generated. The participant was provided the last opportunity to review all projects and concepts and to speak of any concepts left out. In total, 19 interviews yielded 151 raw concepts. The average number of construct per person is 8, with a minimum of 4 and a maximum of 13 (*SD* = 2.53).

(3) Linking Elements with Concepts**:** Having elicited the elements and concepts above, I asked the participant to link concepts with elements in a grid to understand how the participant actually uses the concepts to interpret projects. A grid consisting of a number of rows and columns was created, with the contrast ISD concepts generated by participants on the two sides of rows and elements in columns. Then, I asked participants to rate elements on concepts.

I chose the 9-point scale because the odd number provides a reference point and the degree of freedom to rank eight elements provided in this study (Hunter & Beck, 2000). Table 4 shows an example of the full repertory gird, where 1 relates to the construct pole on the left-hand side of the grid and a rating of 9 relates to the pole on the right-hand side. If the participant sees a project with exactly the same characteristics of the left pole, he/she gives a rating of 1; If the participant sees a project with exactly the same characteristics of the right pole, he/she gives a rating of 9. Appendix D includes a repertory grid of each participant.

Table 4 Sample of Completed Repertory Grid

Similar (1)	Project 1	Project 2	Project 3	Ideal	Project 4	Project 5	Project 6	Suboptimal (Unsuccessful)	Different (9)
User involvement and participation	4	1	1	2	7	7	1	9	A lack of interaction between users and developers
Focus on people	4	4	2	2	6	5	3	8	Focus on process
Continual changes	5	6	2	1	3	4	2	7	Freezing the

in design									design specification after small number of iterations
Continuous integration	6	7	3	2	5	5	1	8	Integration in the end of the project

Since the purpose of this study is to identify the content of ISDP mental models, I did not further analyze the ratings. The rationale of the ratings used in this study was to allow participants to reflect upon different projects when they rated each project, thereby generating more concepts.

Coding ISD Concepts. Concepts are themselves organized in hierarchies, with more abstract concepts at higher levels and more specific concepts at bottom levels (Kelly, 1955; Kunda, 1999). Lower level concepts inherit attributes, characteristics, principles, and core values from higher level concepts. The hierarchy of mental models makes our world manageable.

To further clarify the ISDP mental model for comparison and analysis, it is important to make the terms with similar/identical meanings consistent and classify the raw constructs into an appropriate level. After each interview, I used fundamental ISD concepts and their definitions (see Appendix H) as a coding guideline. Then, each raw concept was assigned with one or more fundamental ISD concepts when it shares the properties of fundamental ISD concepts. Coding was based on the meaning of raw concept and interview notes. For instance, the “documentation for approval vs. documentation based on needs” (participant #5) is coded as the “promote simplicity” concept because the interviewee considered extra documentation as wastes and should be removed.

Raw concepts frequently share properties with multiple fundamental concepts. This is common because concepts are often enmeshed in an individual’s mind. Some people are unable to separate one concept from another at the detailed level. When participant #17, for example, proposed the concept “Source code control that includes version control tools and code review processes vs. a lack of source code control", he suggested that defined processes are needed to ensure development efficiency and software quality. The raw concept was encoded as "Close control over software development processes and procedures" and "Defect detection" because code review processes help detect defects before product releases. The raw concept is also associated with the "Tool/software support" concept because, without version control tool, managing source code from multiple programmers would be extremely difficult. The coding results are summarized in Appendix E.

Point of Saturation. Sampling was terminated when the collection of new data did not shed further light on the issue under investigation. The point is called point of saturation or point of redundancy (D. J. Armstrong, 2005). The interview stopped at the 19th interview after new concepts ceased emerging in three consecutive interviews. The point of saturation is consistent with prior research, which was achieved between 15 and 25 interviews (F.B. Tan & Hunter, 2002). Figure 7 shows new concepts that were obtained in each interview.

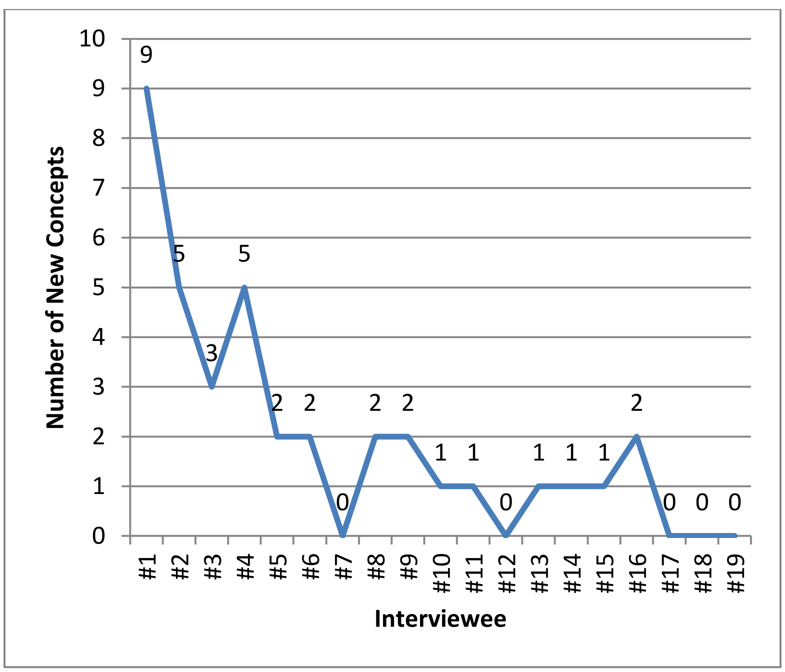


Figure 7 New Concepts Generated from Each Interview

Validity and Reliability. Given that concepts were elicited from ISD experts and were triangulated with a thorough review of the domain in question, the face, content, and construct validity of the ISDP mental model construct should be high. With regard to reliability, I may not able to show reliability in a statistical way (e.g., internal consistency) because IS experts think of different ISD projects (i.e., elicited elements rather than supplied elements), which are incomparable. Yet, arguably the ISD concepts should be all relevant to the topic of interest: ISD concepts about the management of ISD tasks and people. In this study, I did not examine test-retest reliability but prior research has shown that constructs/concepts extracted from the RGT are relatively stable (Wright, 2008).

3.4.2 Results of Study 1a

The main purpose of the content analysis and repertory grid interviews is to define the content of project members' ISDP mental models. I compared concepts generated from repertory grid interviews with concepts extracted from the content analysis. Two concepts found in the content analysis were not mentioned by ISD experts (See the frequencies of the concepts mentioned by the experts in Appendix I). They are "end users' welfare is the major concern of IS development" and "reflect on improvement at regular interval".

The concept of "end users' welfare is the major concern of IS development" comes from ISD methodologies with a socio-technical perspective, such as Effective Technical and Human Implementation of Computer-based Systems (ETHICS) (Mumford, 1995) and Multiviews (Avison et al., 1998; Avison & Wood-Harper, 1990). It considers what and how IT-enabled organizational changes may influence employees' job satisfaction and welfare. From an ethical perspective, this issue should be considered during ISD. ISD professionals who possess this concept would include users in the decision-making process and work together to find ways of improving their job satisfaction (Mumford, 1983). These ISD professionals act not only as project managers, analysts, and programmers but also as change agents. They should, along with other necessary facilitators, reconcile conflicting interests of organizational efficiency and job satisfaction among managers, subordinates, and information systems. This concept has been implemented in some organizations (e.g., Adman & Warren, 2000). However, social and human concerns are not major ethos of ISD implementation (Iivari et al., 2000). It is not surprising that the "end users' welfare is the major concern of IS development" concept did not emerge from our interviews.

However, it is unexpected that no participant came up with the "reflect on improvement at regular interval" concept. Consideration of continuous improvement of the software development process has become prominent over the last two decades. This is largely influenced by numerous successful cases in manufacturing. Relevant frameworks and methodologies include the Capability Maturity Model (CMM) framework (SEI, 1994), Six Sigma, IT Infrastructure Library (ITIL), and Lean development. As common in these frameworks and methodologies, process improvement champions are needed and cyclic assessment and reviews should be set up. Similar ideas can also be seen in recent Agile methodology, such as Scrum. The sprint retrospective meeting is one of the work practices in which a project team spends a small amount of time on a retrospective meeting for every week during the sprint. Basically, the meeting covers questions, such as "what went well during the sprint?", "what would we like to change", and "how can we implement that change" (Schwaber & Beedle, 2001). A recent survey shows that, among a variety of Agile practices, the retrospective meeting is effective and easy to learn (Ambler & Vizdos, 2009). It is surprising that this concept was not mentioned in our interviews, especially when some of the projects were claimed to follow Agile methodologies. One potential explanation is that the notion of reflection on improvement has been embedded in other concepts, such as promote simplicity and continuous attention to technical excellence. Also, one participant suggested that they did not single out some concepts during the elicitation processes because these concepts are too essential and fundamental to be mentioned. Neither explanation would discount its importance, considering its strong support from the literature. To ensure the comprehensiveness of the content domain, both concepts remain in the pool of concepts.

Four new concepts emerge from repertory grid interviews. They are "innovative design", "identify IT/business strategy and align projects with IT/business strategy", "leverage industry standards or best practices for IS projects", and "disciplined change evaluation and management". A missed identification of these concepts could be due to various reasons. They may not be predominant concepts, such as "innovative design". Also, these concepts might have been overlooked because they come from different streams of literature; for instance, "identify IT/business strategy and align projects with IT/business strategy" is primarily discussed in strategy and organization governance literature. Moreover, some important concepts were ignored during the coding process because of the blind spots of my mental model. Take disciplined change management for example, I considered the concept that has been emphasized in PMBOK a subordinate concept of close control over software development processes and procedures. However, participants suggested the change management goes beyond process control and includes evaluation of changes and management of relevant stakeholders. I briefly describe each of these new concepts below.

(a) *Innovative Design*: Innovative design considers multiple solutions at the outset of a project (with agreed upon constraints in mind), assesses their feasibility, and eventually converges on one final solution. Participant #3, for example, adopted this concept to develop a SaaS platform, creating three versions, each with different layouts and interactions. Following testing with target users, one version was chosen and further refined by incorporating design elements from other versions. This concept originated from the manufacturing industry - specifically, TOYOTA - and is also classified as set-based concurrent engineering (Sobek, Ward, & Liker, 1999). Unlike other automotive companies, TOYOTA invests considerable time to explore and examine multiple design possibilities, recognizing that once design decisions are made, they are expensive to change. Different design parties (e.g., usability engineers and architects) are asked to propose, develop, and share various solutions and work closely together with users to narrow down the list of proposed designs. This approach avoids premature design, while stimulating creative solutions (Baines, Lightfoot, Williams, & Greenough, 2006). It differs substantially from other design approaches, such as plan-driven and iterative/incremental process models, which focus on simply improving upon a single solution and are limiting because multiple approaches are never attempted. The concept of innovative design has been suggested by Poppendieck and Poppendieck (2003) for software development: it can be used to convey ambiguous requirements, choose appropriate technologies, and design creative user interfaces. Although innovative design yields good returns for software products in the long run, the upfront cost is higher than that of other approaches, which may deter adoption of this concept.

(b) *Identify IT/business strategy and align projects with IT/business strategy*: Information systems selected for development should support an organization's business and IT strategy in order to impart the organization's values. IS project alignment ensures that project deliverables meet project objectives, as well as IT/business strategy (Jenkin & Chan, 2009). For example, participant #9 mentioned his organization paid attention to business objectives throughout projects. In such cases, project teams are aware of their business objectives and constantly communicate with users to confirm the projects are on the right track. Therefore, I defined the concept as "Identify IT/business strategy and align projects with IT/business strategy".

This concept is related to "Well-defined project charter and project plan that project stakeholders can understand" as it is essential to state business objectives in a project charter (Project Management Institute, 2008). However, the concept also suggests that discrepancies between a business' objectives and actual project deliverables are regularly examined, which increases the likelihood that desired business outcomes are ultimately achieved. Moreover, this new concept differs from, yet is related to, the concept of “Management and control via metrics”. Project alignment requires measurable business objectives, both to recognize these discrepancies and to find ways to correct them.

(c) *Leverage industry standards or best practices for IS projects*: Participants #9, #14, and #17 suggested that adopting or tailoring formalized industry standards can improve software delivery, in such areas as software platform (e.g., J2EE), enterprise architecture (e.g., COBIT, ITIL), and software development (e.g., CMMI, Rational Unified Process). I defined the concept accordingly as “Leverage industry standards or best practices for IS projects”. Industry standards, or best practices, prescribe proven rules and procedures for software development processes that enrich a project team’s capacity to deal with complicated problems. Nevertheless, rigid adherence to industry standards without considering the project context could harm performance (Nidumolu & Subramani, 2003) - appropriate customization is key to maximize the benefits of industry standards or best practices. This concept is prevalent in the ISD community; however, in this study, my intent during concept encoding was to focus on specific practices of ISDP management, and consequently, I missed including this overarching concept.

(d) *Disciplined change evaluation and management*: Participant #19 indicated that changes in funding, timeline, human resources, and scope are bound to have widespread ramifications for a project and should be managed. Initially, we considered placing the concept under the “Close control over software development processes and procedures” concept, which refers to a sequence of steps that regulate how people should behave during software development. Change management, or configuration management, defines rules when project teams encounter changes; accordingly, it can be considered as one such control mechanism. However, I realized that participant #19’s concept was not limited to merely control processes that ensure all changes for work products and associated plans are documented and updated as necessary; rather, she believed that changes should be well controlled even before they are implemented. That is, all changes should be in agreement with the business contract and carefully assessed to understand how they affect the project. After further analysis, I labeled this concept as "Disciplined change evaluation and management".

Including these four new concepts, the number of ISD concepts is 40 in total. The list of 40, including the name and definitions of concepts, was sent to 19 ISD experts for validating clarity of wordings, accuracy of description, and completeness of a list. Twelve out of nineteen participants replied to the follow-up questionnaires. Their responses on the completeness of the list of 40 and definitions of each concept indicate the comprehensiveness of the list (5-point Likert scale, where 1 denotes not at all and 5 denotes very much. *mean* = 4.2, *SD* = 0.63) and clarity of definitions (5-point Likert scale, where 1 denotes not at all and 5 denotes very much. *mean* = 4.2, *SD* = 0.42, respectively). Also, I asked if they could provide any other new concepts. Few respondents provided additional concepts. Three more concepts came up but after further discussion with respondents, they agreed these concepts have been covered by the list of 40. First, participant #19 suggested that a project should have clear project closure to avoid further changes and reflect upon on lessons learned. I considered a project closure is a phase of software development rather a concept. For the effects of clear project closure, avoidance of changes should belong to "disciplined change evaluation and management"; reflection upon lessons learned should be related to "project knowledge management" and "reflect on improvement at regular intervals". Second, participant #19 suggested two work practices: rapid prototyping for experimenting ideas and validating assumptions and risks of their undertaking; using "Spikes" (a practice of Scrum) in between sprints to examine new ideas and assess their risks. Both concepts can be categorized into "Innovative design" and "Explicit recognition and management of risk".

In sum, fundamental ISD concepts derived from ISD methodologies approximate the content derived from ISD professionals. It should be noted that this does not suggest that individuals doctrinally follow ISD methodologies. On the contrary, people often practices differently from what ISD methodologies preach (B. Fitzgerald, 1997). In the next study, the structures of individual mental models demonstrate variances and complexities in which concepts should work together.

3.4.3 Study 1b: Analyzing the Structure

In light of the list of 40 ISD concepts defined previously, I explore the underlying arrangement of ISD project knowledge structure by conducting a survey with ISD professionals where the pairwise rating technique was employed.

3.4.3.1 Elicitation of the Structure Using the Pairwise Rating Technique

Among a number of quantitative elicitation techniques, such as visual card sorting and ordered tree technique, I adopted the pairwise rating technique, which has advantage over others in terms of its ability to capture complex interrelationships between concepts (Langan-Fox et al., 2000). Moreover, the technique is time efficient (105 pairs in 15 minutes, Tossell, Smith, & Schvaneveldt, 2009)[16] and requires little reading or writing. Despite its strength, 780 pairwise comparisons (40*39/2) based on 40 concepts can be a time-consuming task and lead to fatigue. Therefore, I chose a hybrid approach to deal with a large scale dataset as suggested by Markoczy and Goldberg (1995). Only a partial list of concepts (the most salient ones) was selected in order to reduce respondents' efforts to link the relationships between concepts.

Respondents were presented with 40 concepts with definitions and accompanying example work practices. Then, they were asked to choose between 5 and 15 concepts that they consider most important

[16] Experts take a shorter amount of time (13.8 minutes) whereas novices need longer time to complete comparisons (16.2 minutes) (Tossell, Smith, & Schvaneveldt, 2009).

into managing ISD projects successfully (see Figure 8). The reasons for the limitation of 15 concepts are practical (105 pair comparisons) and are based on the results of repertory grid interviews. The maximum number of concepts that was generated from 19 ISD experts is 13. I assume that the number 15 should capture a significant part of mental models.

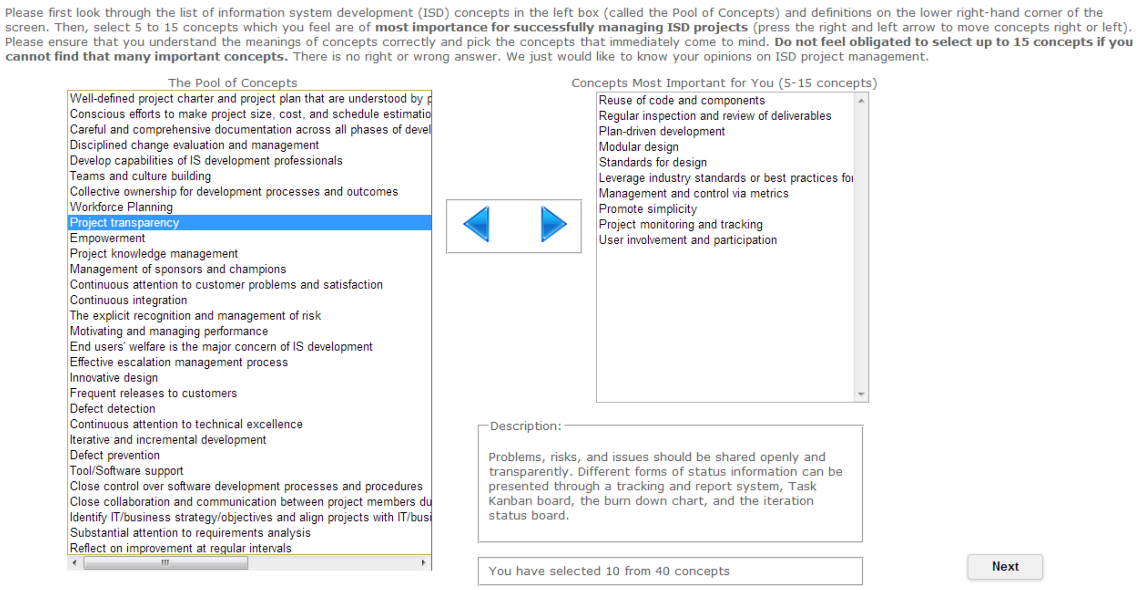


Figure 8 A Screenshot of Concept Selection

Then, based on the concepts respondents selected, they were asked to consider to what extent each pair is related to one another on a 9-point scale[17] from 1 to 9, where 1 denotes unrelated and 9 denotes highly related, in a matrix format (see Figure 9). The sequence of items in the matrix was randomly generated to avoid the order effect (Roberson & Sundstrom, 1990). Appendix K lists the

[17] In the pilot test with 31 ISD professionals with 17 years of ISD experience, I found that the distribution of responses on the -4 to +4 scale is largely skewed. Only 75 out of 1,118 responses fell between -4 and -1. This may be because all concepts are positive in the way to help project management. Respondents hardly consider negative relationships. The skewed data does not affect the robustness of results because Multidimensional Scaling (MDS) is not an inference statistical method and is free of the assumption of normal distribution. The issue is the smaller range (1 to 4) may not provide sufficient options for respondents to discriminate the relatedness. We decide to change the -4 to +4 scale to the 1 (not at all related) to 9 (highly related) scale.

frequency of concepts selected by respondents. Each concept has been chose by at least 10 respondents. This suggests that the provided list captures the interests of the participating ISD professionals.

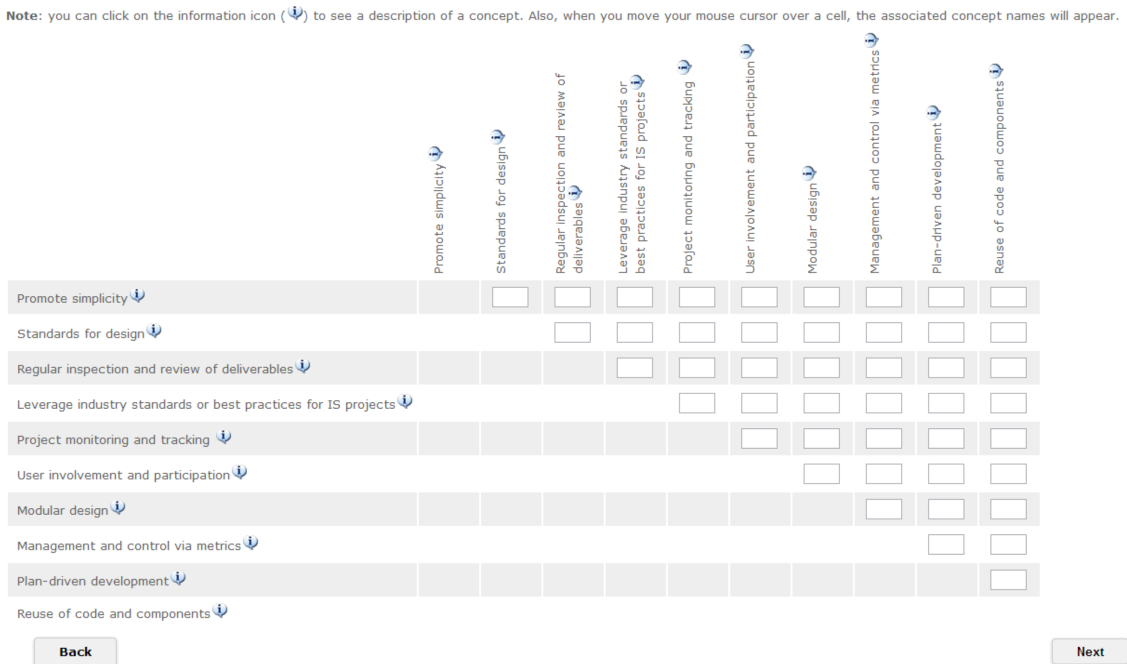


Figure 9 A Screenshot of Concept Comparisons

Participant. One hundred and seven respondents from online professional groups (e.g. Agile CMMI group on Linkedin), AIS Listserv, online subject panels (e.g., Amazon Turks) took part in the online survey. The total valid response was 95. Respondents have an average of 11.92 years' experience in ISD (*SD* = 9.33) and have expertise in programming (*n* = 64), system analysis (*n* = 55), system architecture (*n* = 42), business analysis (*n* = 41), testing (*n* = 55), and project management (*n* = 58)[18]. They are primarily male (69.5%), between the age of 25 and 34 (38.9%), and have college degree or above (80%).

3.4.3.2 Analysis

The objective of study 1b is to understand how ISD professionals perceive the relatedness of ISD concepts and therefore uncover evaluative beliefs. I employed Multidimensional Scaling (MDS) to

[18] Respondents were asked to rate their confidence in different ISD areas using the 5-point Likert scale (1: Not at all; 5: Extremely). We consider the rating greater than and equal to 4 as indicative of expertise in the specific area.

portray the knowledge structure of ISD professionals. This method is ideally suited for our purpose because it enables one to locate ISD concepts in a spatial configuration and illuminates the underlying dimensions of concepts that ISD professionals rely on to understand ISD knowledge. The basic idea of MDS is to find a configuration in a n-dimensional space, in which the distances between objects (i.e., concepts, in this study) in the space approximate the original dissimilarities (Cox & Cox, 2000). For more discussion about MDS, please refer to Borg and Groenen (2005) and Borg et al. (2013).

The source of data for MDS is a proximity matrix, which represents relatedness of concepts. The proximity data can be obtained and analyzed in two ways by MDS (Hair, Black, Babin, & Anderson, 2009). First, aggregate analysis averages the data of individual matrices and represents an aggregate spatial configuration. Second, combined analysis attempts to find an aggregated spatial configuration that fits each individual matrix as precisely as possible. The combined analysis approach not only provides aggregated results but also displays individual matrices. However, it can only be used if respondents rate the same objects. In this study, comparing 40 concepts is a daunting task, which requires 780 paired comparisons. Therefore, I only asked respondents to rate the concepts that they considered important (15 concepts at most). Aggregated analysis therefore is an appropriate choice.

Respondents completed 4,848 comparisons in total. I averaged ratings for each paired comparison. A non-metric MDS with two-dimensions was conducted to represent the knowledge structure of ISD professionals based on both statistical and theoretical rationale. First, the Shepard diagram is used to determine the appropriateness of the solution. The Shepard diagram is a scatter plot of dissimilarity between the observed distance (X axis where represents respondents rating along 1 to 9) and predicted distance (Y axis). When the points on a scatter plot fit a regression line, the n-dimensional spatial configuration represents the data well (Borg & Groenen, 2005). The regression line shows some steps because a monotonic regression is used for the non-metric MDS. The Shepard diagram did not identify significant anomalies as shown in Figure 10.

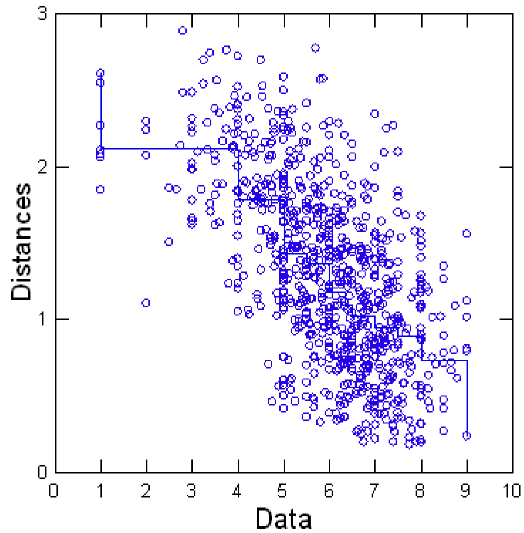


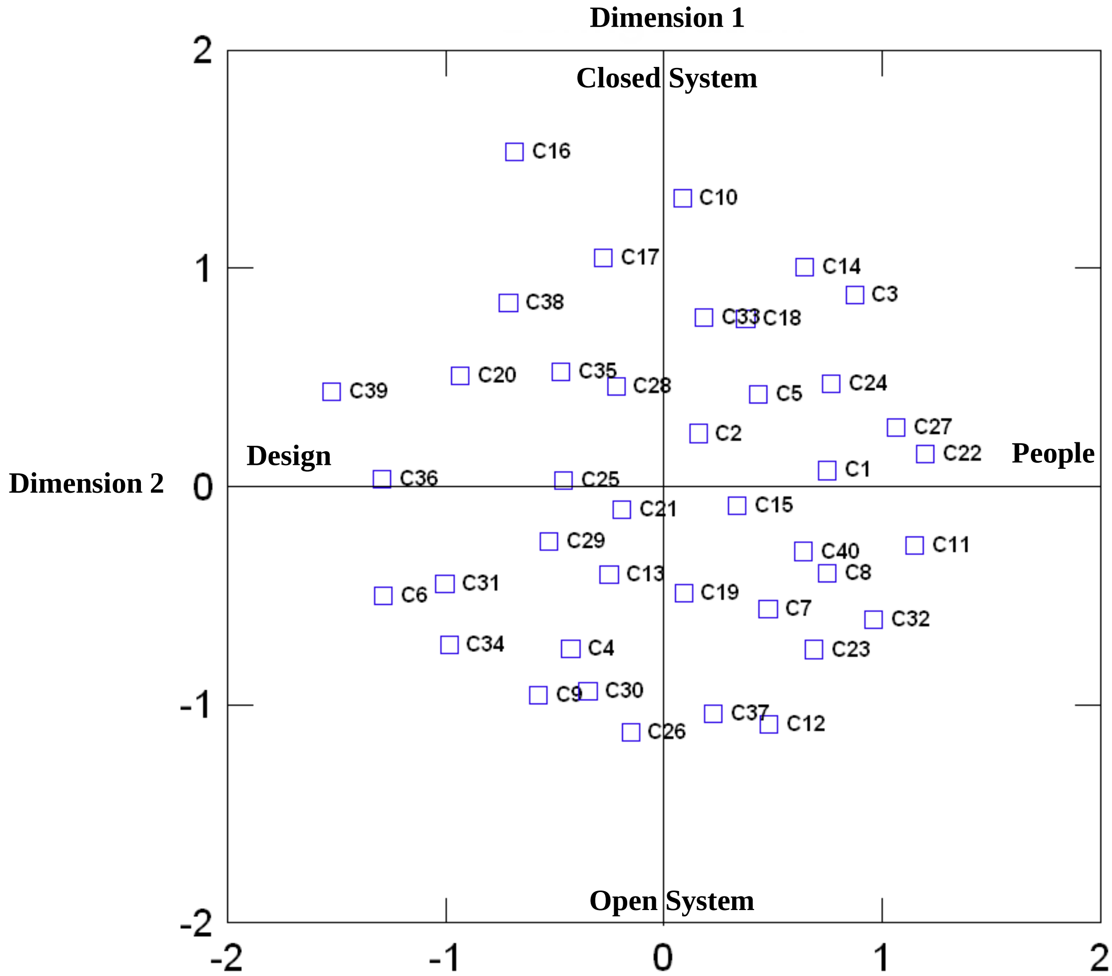
Figure 10 Shepard Diagram of the Two-dimensional Solution

Second, I calculated the stress index (i.e., Stress-1) (Kruskal, 1964), which assesses the badness-of-fit between predicted spatial configuration and the original data. The stress index varies between 0 and 1, where 0 indicates perfect fit and 1 indicates significant misfit. The stress index is 0.31 for the 2-dimensional solution. By Kruskal's criterion, this would not be considered a good fit (< .20). However, the criterion is a rule of thumb and does not take the number of objects (ISD concepts in this study) into account. When the number of objects is small vis-à-vis the number of dimensionalities, the stress values are generally better. It is suggested that if the number of objects is much larger than the number of dimensionalities (e.g., more than 10 times larger), a higher stress value is acceptable (Borg & Groenen, 2005). In this study, the number of objects is 20 times larger than dimensions. Borg and his colleagues (2013) further suggested comparing the MDS solution with a solution generated from random data. A fair solution should have a stress value clearly greater (e.g., two standard deviations) than the stress value expected for random data. According to Spence and Ogilvie (1973), for 40 objects in 2 dimensions, Stress-1 value expected from random data is 0.37, with a standard deviation around 0.005. The observed Stress-1, 0.31, is clearly smaller than the expected random Stress-1.

From the statistical point of view, the observed data can be presented in two or more dimensions. However, considering parsimoniousness and interpretability (M. L. Davidson, 1983), I chose the 2-dimension solution to interpret how 40 concepts are interrelated.

3.4.4 Results of Study 1b

Figure 11 illustrates a spatial model with 40 concepts presented in terms of two underlying dimensions. The distances in the space indicate the relatedness between concepts. The greater the distance between concepts on a map, the less relatedness between them. The figure represents a generic mental model that covers a variety of management principles that help individuals in response to a chaotic project environment. In other words, the figure represents the world of ISD project management as perceived by ISD professionals. In order to gain a better understanding on the underlying structure and examine if three evaluative beliefs suggested by Kumar and Bjorn-Andersen (1990) exist, I propose to examine the 2-dimension structure by virtue of general systems theory (von Bertalanffy, 1968).



Quadrant II	Quadrant I
C16 Conscious efforts to make project size, cost, and schedule estimation C17 Identify IT/business strategy and align projects with IT/business strategy C38 Develop capabilities of IS development professionals C20 Plan-driven development C39 Innovative design C35 Workforce planning C28 Careful and comprehensive documentation across all phases of development C25 Close control over software development processes and procedures C36 Tool/Software support	C3 Team and culture building C14 Well-defined project charter and project plan that project stakeholders can understand C27 Management of sponsors and champions C22 Close collaboration and communication between project members during development C24 Effective escalation management process C10 Motivating and managing performance C18 Empowerment C33 Project knowledge management C5 Management and control via metrics C2 Project monitoring and tracking C1 Explicit recognition and management of risk
Quadrant III	**Quadrant IV**
C6 Standards for design	C11 User involvement and participation

C34 Modular design C31 Reuse of code and components C9 Defect detection C30 Iterative and incremental development C26 Continuous integration C4 Promote simplicity C29 Leverage industry standards or best practices for IS projects C13 Continuous attention to technical excellence C21 Regular inspection and review of deliverables	C32 Reflect on improvement at regular intervals C40 Disciplined change evaluation and management C8 End users' welfare is the major concern of IS development C23 Continuous attention to customer problems and satisfaction C7 Collective ownership for development processes and outcomes C12 Defect prevention C37 Frequent releases to customers C19 Substantial attention to requirements analysis C15 Project transparency

Figure 11 Two-dimensional Configuration of the ISDP Mental Models

General systems theory suggests that any kinds of systems (e.g., organizations) can be explained by a set of principles. A system is comprised of four elements: objects (e.g., people), attributes (e.g., roles and skills of people), relationships among objects, and environment where a system dwells. Several generic principles can be used to understand systems. For instances, objects in a system are interrelated and can collectively contribute to synergic influences on a whole system (wholeness); multiple sub-systems are embedded hierarchically in a system (hierarchy); a system can be considered as open or closed (openness). The theory has been adapted to study organizations (Ashmos & Huber, 1987; Kast & Rosenzweig, 1972). I maintain that the theory offers a lens to understand how ISD professionals make sense of ISD project knowledge. Specifically, general systems theory provides a framework to understand how an ISD project interacts with its environment (Dimension 1) and People/Design systems (Dimension 2).

Dimension 1. Relationships between concepts along the first dimension suggested a label reflecting an orientation toward management of internal operations vs. an orientation toward management of external environmental changes. The opposing orientations along Dimension 1 reflect another assumption stemming from general systems theory, which classifies organizations as open systems or closed systems. Concepts that make different assumptions are located far apart in the space of mental models.

When the closed system is assumed, it is believed that the software development tasks are more predictable and less affected by environmental factors. Most of concepts in this end aim to develop internal IS capabilities (Ravichandran & Lertwongsatien, 2005) to reduce uncertainty and to create buffers for changes. For example, through planning and control, such as estimation procedures and techniques (C16), project charters and plans (C14), strategic alignment and planning (C17), and human resource planning (C38), projects can be better managed. Moreover, individual productivity can be enhanced and development effectiveness increases.

In contrast, the open-systems assumption accepts the nature of complexity in a project environment. To process complicated and equivocal environmental demands, a project team should pay attention to information coming from customers (C23), closely interact with customers (C11), and leverage different technical practices, such as continuous integration (C26), frequent releases (C37), iterative and increment procedures (C30), and test-driven development (C12), to react to changes coming from interaction. Reflection on changes and improvement (C32) is also a vital element in maintaining adaptability toward the complex environment.

Dimension 2. Relationships between concepts along the second dimension suggested a label reflecting the design vs. people regarding the management of ISD. According to the hierarchy principle of general systems theory, a project can be regarded as a type of sub-systems belonging to the higher-level organization system. A project itself is also a system that contains lower-level systems (e.g., humans). In order to achieve synergetic benefits, people employ different ways to change the properties of different levels of systems in organizations. The "ways" manifest themselves as concepts in this study and can be separated into two aspects: design and people.

The design aspect encompasses design of organization systems and software systems. At the organizational level, projects are more likely to meet their own objectives and organizational strategies through tweaking and designing processes, structures, and practices. The design aspect has its root from rather traditional management logics, such as the idea of scientific management that focuses on the careful design of the process and structure (Taylor, 1911), but it has blended modern management principles, such as quality management, concurrent engineering, and process reengineering. As shown in Figure 11, these concepts aim to change organizational systems so as to improve project effectiveness, such as development processes for innovation (C39), technology-supported project environment (C36), and systematic development processes (C20). At the operational level, the major concern is software systems - the core of an ISD project. Drawing mainly from software engineering, concepts in this subcategory strive to establish effective processes of development, such as design standards (C6), code and component reuse (C31), and modular design (C34), in order to make components of software systems work smoothly.

The other end of Dimension 2 represents another stream of management logic that focuses on human systems, specifically people relations (Mayo, 1933). The value of human resources and the welfare of the people are at the center of this school of thought. Concepts in this end tend to satisfy people's needs and satisfaction to maximize benefits of human capital. Developers care about welfare of users (C8). To address users' problems (C23), they involve users into the development processes (C11). Besides customers, relationships among project members (C3) and relationships with sponsors and champions (C27) are also valued. Close communication and collaboration is the key to make human systems work (C22). It should be noted the management logic of people is not solely built upon normative control and also requires a certain degree of behavioral and outcome control. For instance, project transparency (C15) enhances behavioral observability and therefore reduces risks of social exchange among project members. This is supported by previous research findings that effective use of Agile methodologies requires outcome control (Maruping, Venkatesh, & Agarwal, 2009). Performance management (C10) that collects the metrics (C5) about individual and team works allows project managers to trust their developers even when project managers have little knowledge about programming (Kirsch, 1996). Relatively close distances between some seemingly irrelevant concepts reflect how ISD professionals resolve dilemma during interaction – showing respect for people while preventing anomie.

Quadrants. Two-dimensional configuration suggested that ISD concepts can be classified into four categories and constitute four types of evaluative beliefs. Figure 11 exhibits four quadrants and their underlying concepts.

The first quadrant contains concepts pertaining people and closed system thinking. People who have such a mental model structure tend to encourage involvement of project members and stakeholders (C14, C18, and C27), foster trust and respect among team members (C3), ensure highly motivated teams (C10), and maintain shared understanding among project members and stakeholders (C14, C27, and C33). I refer to this quadrant as the team-oriented beliefs. Figure 12 zooms in on the structure underlying the team-oriented belief.

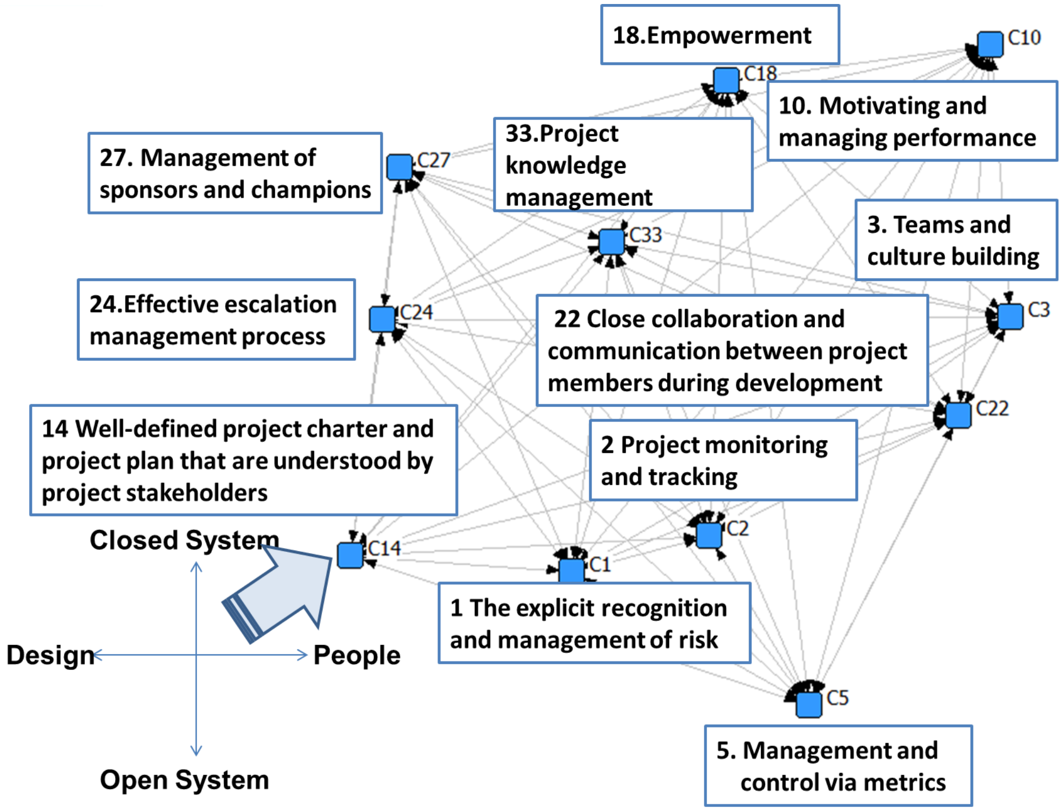


Figure 12 Team-oriented Belief

The second quadrant comprises concepts in the intersection of closed system and design. The focus is on planning and processes. The structure represents not only lower-level project process management (C16, C20, C39) but also enterprise-wide management (C17, C35, C36, C38). The essence of ISDP concepts goes beyond project schedule and cost, and it aims to look outside the context of a project. It is proposed to integrate and standardize resources and processes across projects. The ultimate goal is to obtain strategic business value, which is labeled as the enterprise-oriented belief. Figure 13 zooms in on the structure underlying the enterprise-oriented belief.

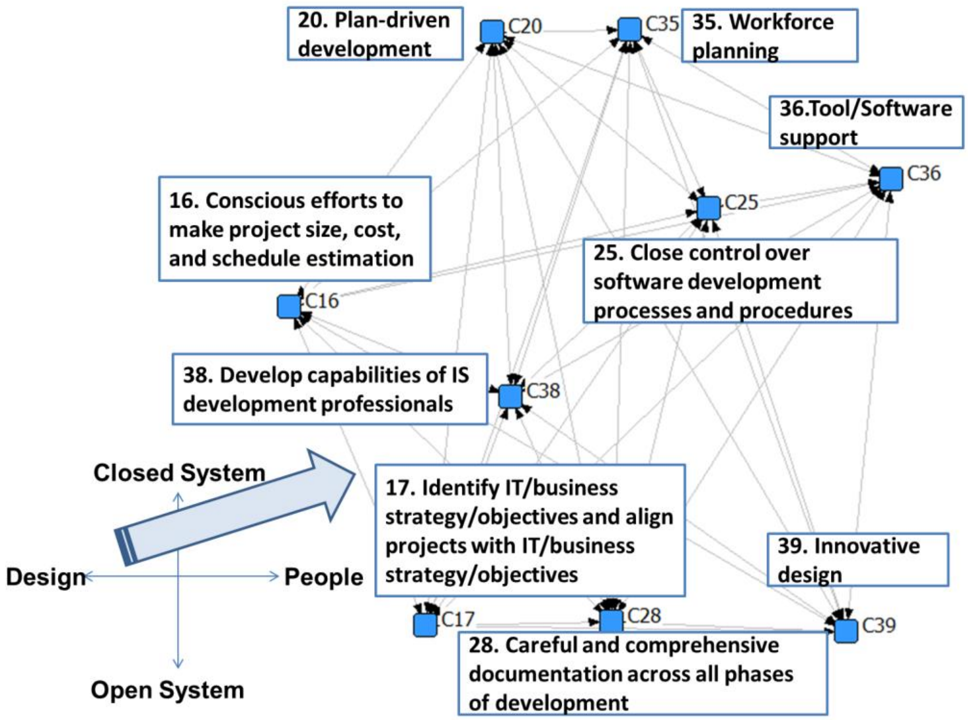


Figure 13 Enterprise-oriented Belief

Concepts in the third quadrant are associated to the creation of software products with high quality. The design of IS aims to respond to volatile external environment. One prominent characteristic of the concepts in this quadrant is pursuing highly adaptable software. The concentration is on developing ease to adapt information systems with high quality. Adaptability can be obtained through modular design (C34), reuse (C31), iterative and incremental development (C30), and design simplicity (C4). Accordingly, the quadrant is referred to as the product-oriented belief. Figure 14 zooms in on the structure underlying the product-oriented belief.

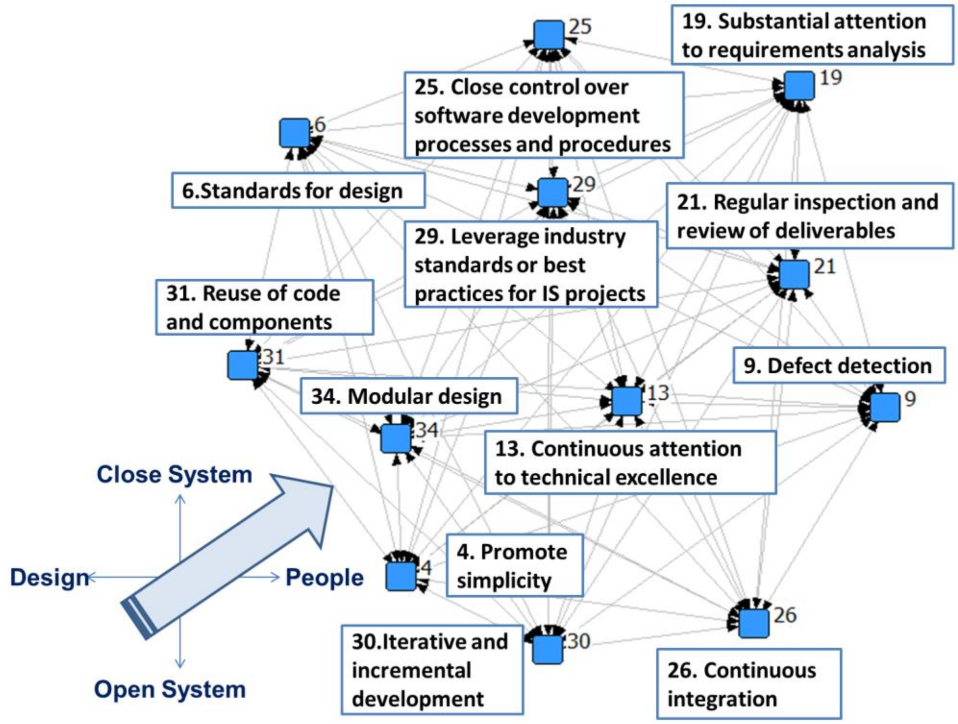


Figure 14 Product-oriented Belief

Although both the first and fourth quadrant are concerned with people, the fourth quadrant puts an emphasis on customers rather than internal project members. Concepts include a formation of partnership where developers and customers closely collaborate (C11) and clear individual responsibilities and accountabilities (C7). To forge effective partnerships, project teams deliver features frequently to meet customers' needs (C23 and C37), which could in turn help customers benefit from the competition in the market. Frequent releases can be in favor of both venders and customers. Yet, frequent releases require good management of evolving processes of software products. Therefore, defect prevention mechanisms (C12) and change management processes (C40) should be in place to support adaptive development. Continuous integration (C26) can also be used to support frequent releases (it is located in the third quadrant but is very close to this quadrant as shown in Figure 11). Given the heavy emphasis on business value that can be brought by/to customers, I label this quadrant as customer-oriented belief. Figure 15 zooms in on the structure underlying the customer-oriented belief.

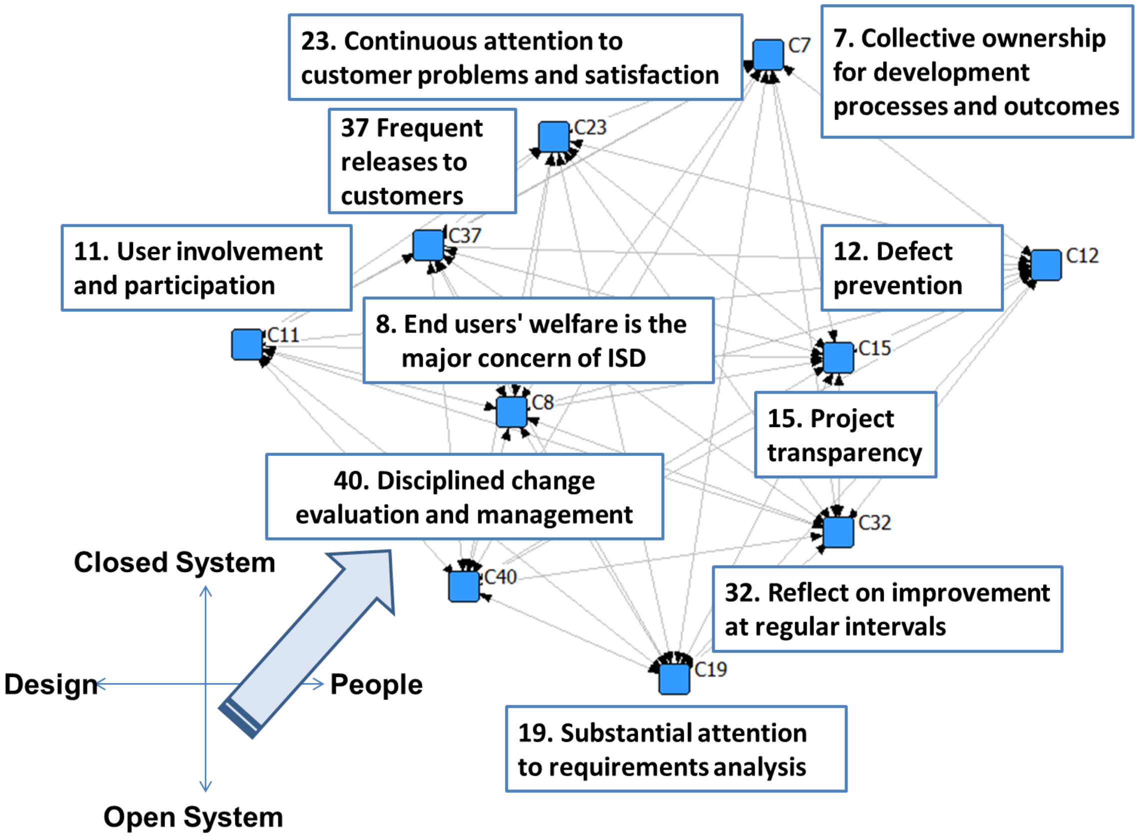


Figure 15 Customer-oriented Belief

3.5 Discussion

Study 1 integrates knowledge accumulated in ISD methodologies with knowledge represented in ISD professionals' mind, thereby empirically laying out different types of evaluative beliefs. The list of 40 concepts contributes to the building block of the ISDP mental model construct. Not surprisingly, I found a high degree of similarity between the content of one's mental model and the concepts I derived from ISD methodologies. The list of 40 reveals what concepts practitioners hold in their minds nowadays.

With regard to the organization of ISD project knowledge, the relatedness between concepts substantiates an understanding of how concepts work together to achieve project effectiveness. The results of study 1b suggested that the management of ISD varies in two dimensions and can be classified into four evaluative beliefs. On the one dimension, I observed the fundamental management principles in terms of design and people. On the other dimension, I recognized two orientations toward the management of ISD in that closed-system thinking focuses on internal organization systems and their effectiveness by means of a variety of strategies, policies, procedures, and practices; open-system thinking accepts the rapidly changing nature of environment and thus believes it is important to maintain interaction between project teams and external entities, such as customers and users, and carry out practices that adapt to changes. The two dimensions guide four foci about the management of ISD: customer, team, enterprise, and product.

This typology extends Kumar and Bjorn-Andersen's (1990) three proposed beliefs. Enterprise-oriented belief is an extension of economic belief. The resources not only need to be well managed but also need to connect to strategic goals. Product-oriented belief is consistent with technical belief. Interestingly, the results suggest that when the socio-political aspect is considered (the people dimension), open-system thinking and closed-system thinking differentiate the socio-political belief into customer-oriented belief and team-oriented belief respectively. ISD professionals who highly value people may possess these two beliefs simultaneously. However, they may set priority for one over another, accordingly influencing their preferences and cause-effect beliefs about ISD project knowledge. Overall, the structure demonstrates four ISD project development beliefs: caring customers to generate financial gains, cultivating teams to accomplish tasks, integrating processes and structures to meet strategic goals, and focusing on technical excellence to ensure software quality. The typology is parallel to the movements of ISD over the last few decades, including, but not limited to, the Agile movement, Lean development, and Enterprise framework.

The discussion so far has addressed the research question in terms of the definition of project members' mental models – the content and structure in particular. The following questions then arise. What does a project member's mental model look like? What can we learn from a project member's mental model? The next two sections provide a post-hoc analysis on individuals' mental models and suggest a procedure to assess evaluative beliefs: a critical component of individual's mental models.

3.5.1 Revisit Methodology and ISDP Mental Model: An Example

In Figure 6, I depicted that project members' mental models are influenced by ISD methodologies through learning and methodology-in-action via directly being exposed to the methodology (formalized instruction or social influence from other project members). In this section, I provide an example to describe the relationship between methodologies and mental models.

By examining respondents who provided the information about the methodology (ies) they practice, I found that one's mental model is not completely organized as methodologies prescribe. Even for people who have systematically learned methodologies, their mental model can be influenced by prior learning or their work in the trenches. More specifically, I observed that mental models are driven by beliefs. This can be illustrated by a mental model of a project leader (F07 - a code to disguise identifying information) who works in the Software sector and has 15 years of ISD experience (see Figure 16). The figure is produced by NetDraw (Borgatti, 2013) using MDS algorithm.

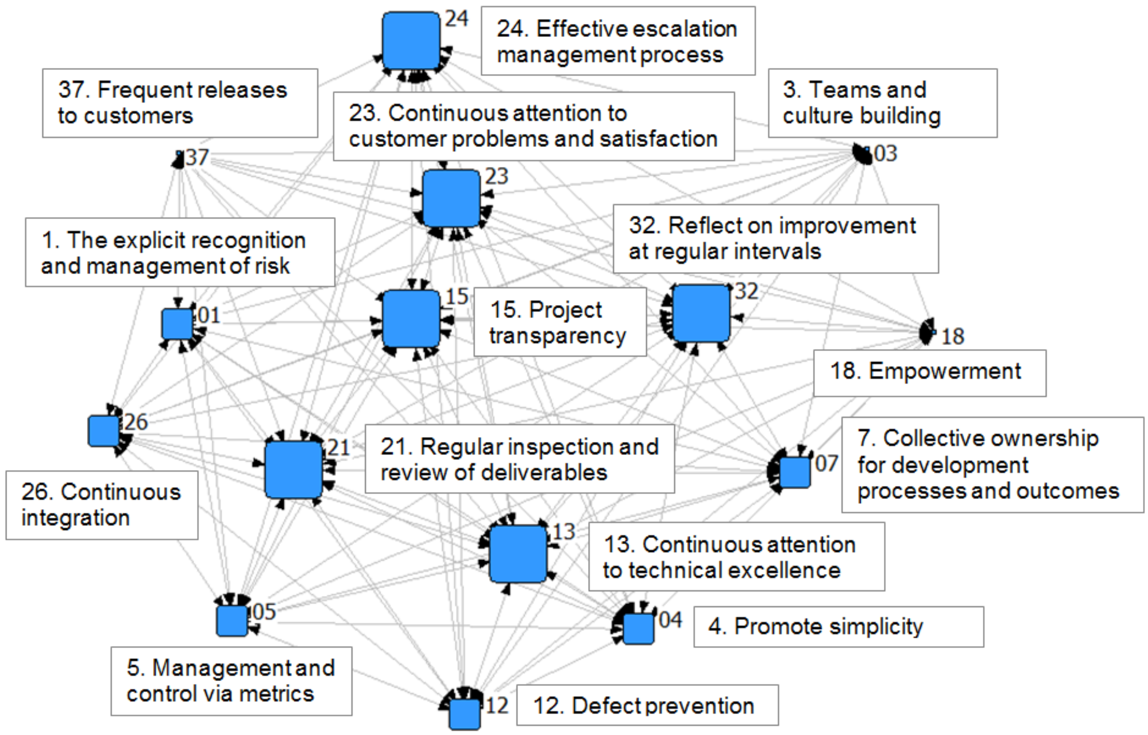


Figure 16 ISDP Mental Model of F07

Each node (round square) represents a concept. The node label is a concept ID. The size of the node indicates the salience of a concept based on the degree centrality of nodes. The premise of the degree centrality index is that when a concept receives more connection from other concepts, the concept is more accessible and is likely to be activated. To calculate the degree centrality, I transformed the continuous relations (the scale of the relatedness is from 1 to 9) to the dichotomous relations with the cutoff value at 5. Please note that all concepts on a map are important to the person because respondents were asked to choose up to 15 most important concepts. Smaller nodes are less related to other major sets of concepts. A solid line between nodes represents a relationship. The longer the distance between concepts on a map, the greater the difference between them.

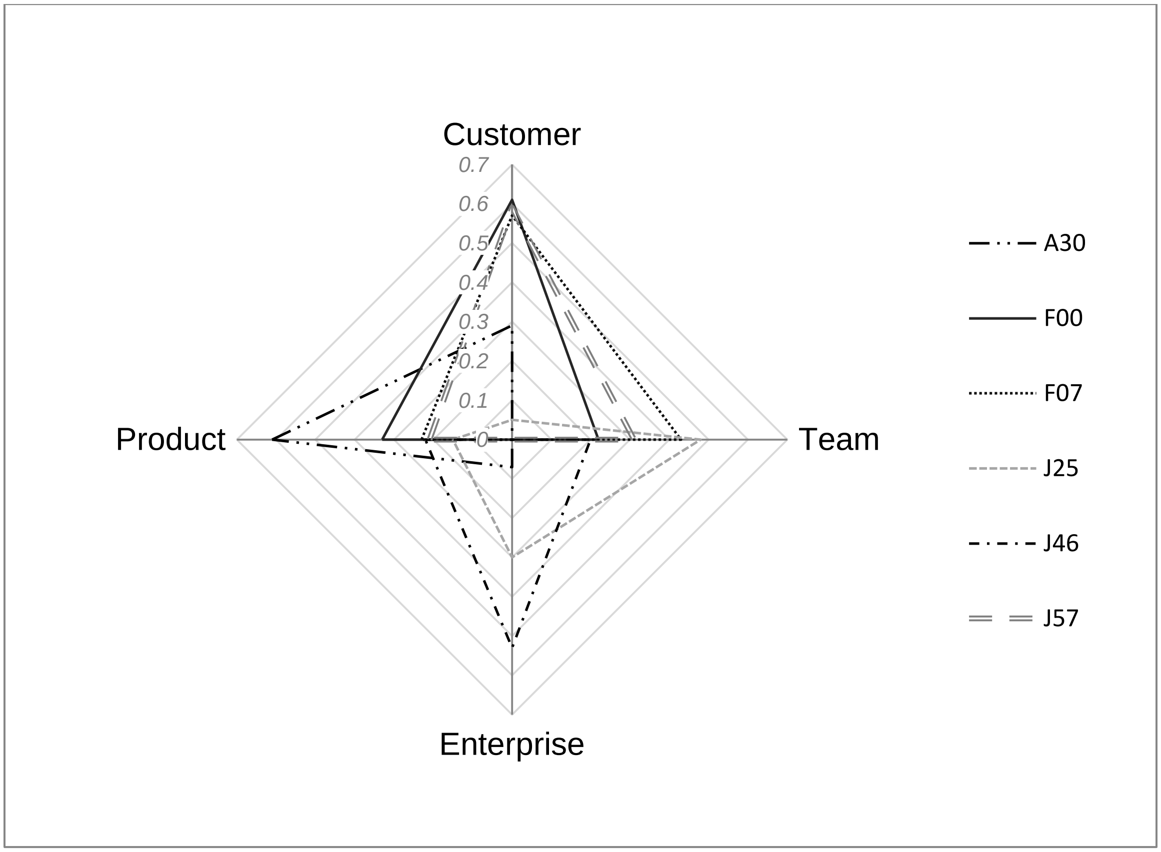
This project leader claimed that he practices Lean Agile (Poppendieck & Poppendieck, 2003). The major principles of Lean Agile apparently exist in his knowledge structure, such as promote simplicity (4), empowerment (18), and management and control via metrics. However, there are some concepts not associated with Lean Agile, such as defect prevention (12), explicit recognition and management of risk (1), and effective escalation management process (24). Moreover, some essential concepts of Lean Agile, such as iterative and incremental development, are left out in his mental models. Differences between his mental models and Lean Agile may not be surprising as a human is not a machine and will not replicate every principle of methodologies. The more important message in this figure is that the mental model conveys more information than it appears - ISD professionals preach and practice certain concepts because they want to obtain certain outcomes. The interpretation of one's evaluative belief is determined by the mental model as a whole rather than focusing on single concept (e.g., promote simplicity (4)). To derive the information underlying the mental model, I propose a profile identification approach.

3.5.2 Uncover a Hidden Message of Mental Models: Assessment Procedure of Belief Systems

In light of the notion of profile deviation (Venkatraman, 1989), this study developed "ideal" profiles for four types of beliefs (customer-oriented, team-oriented, enterprise-oriented, and product-oriented). I used the proximity data of each quadrant to construct a profile of mental models. For concepts located on the boundary of quadrants (i.e., the X and Y axis), I added them into both adjacent quadrants (e.g., concept 25 was deemed an element of the profile for both third and fourth quadrant as shown in Figure 11). Figure 12 to Figure 15 present the prototypical structure of each belief. It should be noted that the prototypical structure does not prescribe the best way of software implementation. Put differently, the adoption of all concepts is not a necessary condition to software product success. The mental models serve as a summarization of concepts alike in terms of their capabilities to achieve similar goals. Take the prototypical structure of product-oriented belief for example, these concepts come from different methodologies, such as object-oriented methodology on iterative and incremental development (30), reuse of code and component (31), and modular design (34); eXtreme Programming on continuous integration (26), iterative and incremental development (30), continuous attention to technical excellence (13), but these concepts share one similar goal: easy to adapt and product quality. Arguably, if people care about product-related outcomes, their mental models should resemble this structure to a certain extent.

To determine one's belief orientation, I calculated differences (deviations) between individual mental models and the four prototypical structures of beliefs using the distance ratio formula that I revised based on Markoczy and Goldberg (1995). The formula considers differences in terms of the selection of concepts and the relationships among common concepts (see Appendix L for formula and an example) and indicates the degree to which individual mental models differ from the prototypical structures. The distance ratio ranges from 0 to 1, where 0 denotes no difference between knowledge structures and 1 denotes completely different. To help interpretation, I transformed the distance ratio to a similarity index by subtracting the distance ratio by one.

As shown in Figure 17, the project leader's (F07) mental model manifests customer-oriented and team-oriented beliefs (the similarity index is 0.57 and 0.43 respectively). This implies that the project leader has a set of salient customer-oriented and team-oriented concepts and puts customers and teams in the very center during project implementation. Interestingly, he relies less on product-oriented concepts and shows no concern for enterprise-oriented concepts. It may worth considering the enterprise-wide concepts, such as IT/business/project alignment (C17), as Lean Agile suggested, in order to maximize benefits of Lean, and an integrated understanding of the overall system, including business processes and IS, is needed (Poppendieck & Poppendieck, 2003).



	Customer	Team	Enterprise	Product	IS/T experience	Job Title/Sector	Expertise
A30	.29	0	.07	.61	32	IT Director/ Manufacturing	Programming and Testing
F00	.61	.22	0	.33	22	Scrum Master/ Government	PM and Testing
F07	.57	.43	0	.23	15	Senior Project Lead/Software	PM and BA
J25	.05	.48	.30	.15	9	Owner of IT company/Software	PM and Testing
J46	0	.20	.53	.22	10	Project Manager/ Software	PM, BA,SA, BArc, SArc, Programming, and Testing
J57	.59	.31	0	.21	29	Project Manager/ Finance and Insurance	PM, BA, Programming, and Testing

Note. *PM = Project Management, BA=Business Analysis, SA = System Analyst, BArch=Business Architecture, SArc=System Architecture.*

Figure 17 Sample Profiles of ISDP Mental Models

3.5.2.1 Diversity of Mental Models

Figure 17 also illustrates diversity of mental models across ISD professionals. Some people's knowledge structure may be dominated by one belief but still showing their attention to other beliefs, such as J57's dominant belief on customer with developing beliefs on team and product. He made the following comments on his mental models.

> *"The top priority I used to determine the project outcome is the client satisfaction. No matter how wonderful of a product IT creates if the clients don't like it or won't use it then the project has failed. I also used ease of testing as the product is being developed as a criterion [to relate the concepts I chose]. Any and all changes will affect the entire project. Therefore testing is the next most important thing to me. It must be unit tested, system tested, user tested, compared back to requirements tested, etc. and must be done every time a change is made. Therefore, modularity, communication between team members, etc. are critical. Also rewarding the team is important as people are people. No one works in a vacuum. Little rewards along the way such as praise or a team outing is needed for morale. People tend to worry about the end result without making sure the process of getting there is acceptable to the team."* (J57)

Some mental models may be aligned with multiple beliefs, such as F07 on customer and team. I am not suggesting that a more diverse mental model is better. However, I speculate that people who possess multiple beliefs are more likely to act as a liaison in a project team. They are able to communicate with people who have different mindsets, thereby contributing to shape shared team mental models (Yang, Kang, & Mason, 2008). This argument warrants future research.

This raises another question: Are ISD professionals' beliefs related to their job roles? There are diverse job roles across respondents. Three primary job role groups are Programmer (n = 26), Project Manager (n = 14), and IS/IT Manager (n = 13). Other roles includes Analyst, Architect, Consultant, IS/T expert (e.g., professors). Given the limited sample size in each group and some missing data regarding job roles, I looked at correlations between four belief orientations and self-reported expertise instead of job roles (5-point Liker Scale, including project management, business analysis, system analysis, business architecture, system architecture, and testing) (n = 87). As shown in Table 5, team-oriented belief becomes stronger when one accumulates more business analysis expertise (r = .22, p < .05) but the belief crumbles as one's expertise in programming develops (r = -.25, p < .05). In a similar vein, product-oriented belief is negatively associated with business analysis expertise (r = -.25, p < .05) and positively associated with programming expertise (r = .23, p < .05). It appears that as ISD professionals gain more knowledge about programming, their product-oriented belief takes precedence over team-orient belief. Conversely, business analysis expertise strengthens team-oriented belief. We should be cautious about the explanation. The weakened belief does not mean that ISD professionals no longer believe in it. Rather, ISD professionals may still adhere to the belief but give priority to another stronger belief. In generally, based on descriptive statistics, there is no statistically significant relationship between job expertise and beliefs. It implies that beliefs may be shaped by other factors, such as individual experience, organizational culture, or national culture (Kankanhalli et al., 2004). This line of thought deserves further investigation because it would be helpful for organizations to understand the antecedents of employees' beliefs related to ISD and for organizations to come up with interventions for aligning employees' beliefs with organizational values.

Table 5 Coorelations between Expertise and Beliefs

	M	SD	1	2	3	4	5	6	7	8	9	10	11
1.PM expertise	3.89	1.14	1.00										
2.BA expertise	3.30	1.30	.73**	1.00									
3.SA expertise	3.66	1.05	.45**	.47**	1.00								
4. BArch expertise	3.05	1.33	.51**	.64**	.53**	1.00							
5. SArch expertise	3.44	1.25	.25*	.25*	.66**	.61**	1.00						
6. Programming expertise	3.93	1.28	-0.03	-0.05	.33**	0.08	.39**	1.00					
7. Testing expertise	3.85	1.12	0.13	0.13	.33**	0.08	.28**	.48**	1.00				
8. Customer-oriented belief	0.15	0.16	0.03	0.19	0.08	0.02	-0.19	-0.09	0.06	1.00			
9. Team-oriented belief	0.17	0.16	0.18	.22*	-0.04	0.08	-0.18	-.25*	-0.07	0.14	1.00		

10. Enterprise-oriented belief	0.16	0.17	0.13	0.00	0.12	-0.04	0.06	0.13	0.18	-.22*	-0.01	1.00	
11. Product-oriented belief	0.19	0.15	-0.20	-.25*	-0.12	-0.21	-0.08	.23*	0.18	.24*	-.21*	0.11	1.00

Note. PM = Project Management, BA=Business Analysis, SA=System Analysis, BArch=Business Architecture, SArc=System Architecture

Note. ** $p < .01$, two tailed. * $p < .05$, two tailed.

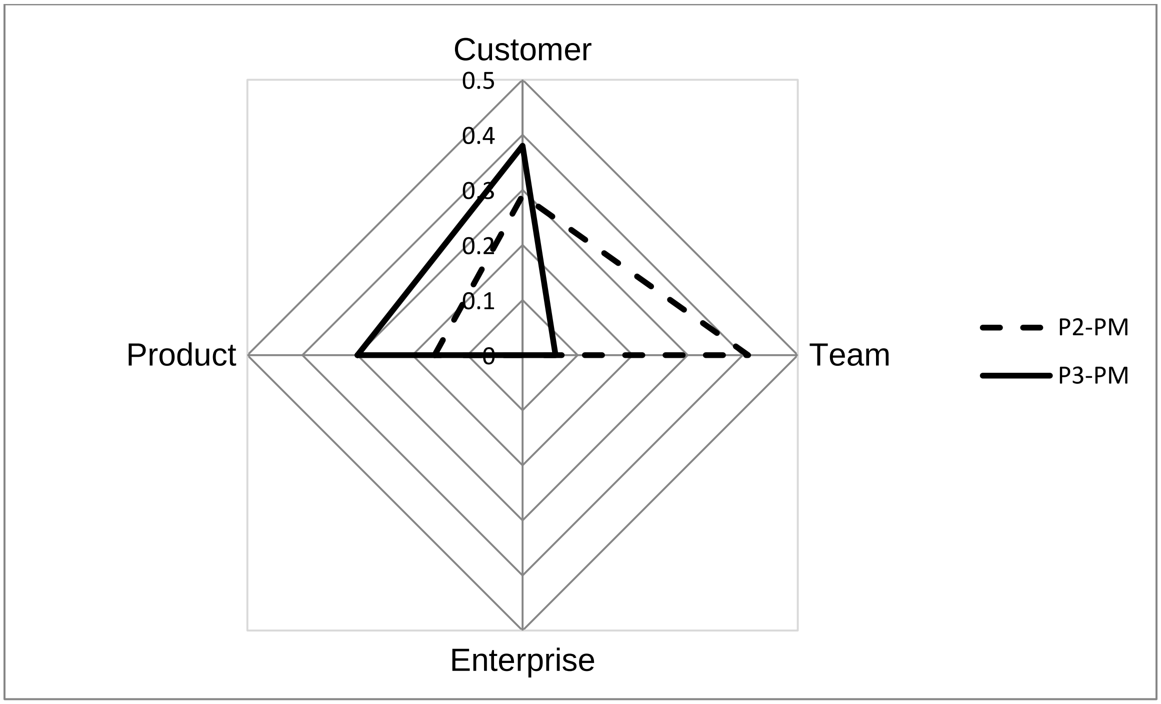
3.5.2.2 Validation of the Assessment.

To ensure the validity of the assessment procedures, I interviewed fourteen ISD professionals, including seven project managers and seven developers, who have approximately 11 years of ISD experience (SD = 6.64) and work in IT consulting (n = 6), software (n = 6), and finance (n = 2) industry (These participants are the same as those in study 2. See Table 6 for the background of participants). They went through the online assessment procedure described in study 2 and they were invited for follow-up interviews (approximately 1 month after the survey).

At the beginning of the interview, I did not show them their mental models and belief orientation. Rather, I asked them to describe their knowledge and beliefs in the management ISD project, such as "*When you work on software development project, which types of outcomes do you think are most important for you?* " and "*How do you achieve these outcomes?*" In the end of the interviews, I demonstrated figures of their mental models and belief orientations (see Appendix O) and asked them to confirm the accuracy of the mental models. The design avoids leading them to think about certain concepts before they answer questions and allows more objective evaluation. All fourteen participants confirmed the accurate representation of metal models in terms of content and association. In terms of beliefs, thirteen participants agreed with the assessment results. For instance, P2-PM (a code to disguise participants' identity) made the comment on the results.

> *"…my boss wants me to focus more on the customer. Because that's my weakness – I am a passionate defender and supporter of my teams and sometimes I need to focus more on my customer. So it will be interesting to reflect more on those and try and figure out how I can try*

and marry the needs of the customer with the needs of the team and try and reflect more often on the customer." (P2-PM, see Figure 18 for her dominant belief)



	Customer	Team	Enterprise	Product
P2-PM	.29	.41	0	.16
P3-PM	.38	.06	0	.30

Figure 18 Examples of Elicited Belief Systems

One case requires extra attention. P3-PM holds beliefs in customer, team, and product and her dominant belief is customer-oriented (see Figure 18). However, according to the interview, team-oriented belief should be her dominant belief. There are two potential explanations. First, she mentioned during the survey she paid much attention to the logical part of project management rather than her strongest beliefs in project management. She might have misread the instruction on the survey. If this is the case, this can be considered a random error. Another explanation is about the limited number of concepts to select from[19]. P3-PM may value other concepts but were not allowed to select them. If the additional concepts and top 10 concepts are weighed similarly, the results may be biased. Future research should take the number of concepts into consideration. To sum up, our validity test shows high accuracy for both knowledge (100%) and belief (92.3%). This should provide sufficient support for the assessment procedures.

3.5.3 Limitations

Several limitations of this study should be noted. First, despite less reading needed for the pairwise comparison, respondents might misunderstand concepts, leading to biased results. I argue that the risk was minimal due to the fact that people chose the concepts that they were well aware of and hence should be able to interpret them appropriately. In addition, definitions were provided in the survey using information icons. Respondents should be able to clarify the meanings of concepts that they felt confused.

Second, I may have been biased to derive and interpret the dimensions of underlying structures. I attempted to offset this potential bias by validating the four evaluative beliefs with independent and blind participants. However, it should be noted that participants in this study have background in business application development. Whether the four beliefs can be generalizable to some other types of information systems, such as mission critical IS, needs further examination.

Third, interpretation should be cautious about the duration of applicability of the results from the assessment procedure. Can the mental models captured at a given point of time accurately represent respondents' mental models after a while? Mental models, particularly for knowledge and belief structure

[19] P3-PM is a participant of Study 2. Participants in Study 2 are limited to select up to 10 most important concepts as compared to Study 1 with up to 15 concepts. Based on the results from Study 1, respondents select 10 concepts on average (*SD* = 3.91). Considering the length of the survey of Study 2, including some other measures, I set the maximum number of selected concepts at 10 to reduce the number of pairwise comparisons from 105 to 45.

discussed in this study, shift slowly in general because people suffer confirmation bias (Wason, 1960) (a strong bias in favor of a preferred or existing beliefs and expectations), dogmatism (Rokeach, 1960) (resistance to change in a person's belief system), or cognitive entrenchment (Dane, 2010) (a high level of stability in one's domain schemas due to increased expertise). However, previous research suggested that people who have certain individual traits, such as flexibility or openness to experience (D. Miller & Toulouse, 1986), or who are novice (Dane, 2010) tend to experience relatively rapid changes in mental models. Future longitudinal research examines the rate of changes in those types of ISD professionals will be beneficial to understand if the interpretation in the present study is still valid.

Lastly, the mental model I developed did not assess the perceived causal relationships between concepts. Instead, I assessed the relatedness between concepts. The causal relationships allow further analyses on the interdependence of concepts. Some concepts may serve as a precondition of other concepts. For instance, building team culture can be a necessary condition to close collaboration and communication. Also, researchers can include concepts of project outcomes required for a specific project (e.g., cost, productivity, schedule, quality) and ask respondents to draw linkages between ISDP concepts and project outcome concepts. The results can shed light on how project members view different ISDP concepts in the context. The reasons of excluding causal relationships in this study are twofold. First, the primary goal of this study is to reveal knowledge and belief structure in general - what fundamental concepts ISD professionals possess, how these concepts are related, what beliefs underlie the association of concepts. This study does not aim to study specific causal relationships between ISDP concepts or how ISDP mental models are applied in a specific project. I will leave these two interesting and worthwhile ideas for future research. Second, from a methodological perspective, causal mapping requires more efforts during the assessment. Given the number of concepts I asked to compare along with other survey measures in the questionnaire, it would be less feasible to expect participants to willingly complete all tasks at once. I would suggest that future research can separate the mental model assessment from other measures to avoid lower quality of responses due to fatigue.

Chapter 4

Study 2: Work Relationships between ISD Project Managers and Developers

Study 2 adopts the ISDP mental model construct developed in study 1 and explores how the interplay of mental models, particularly similarity and understanding, influences work relationships between ISD project managers and developers (see Figure 19). The overarching research question being addressed in study 2 is: *How does the interplay of ISDP mental models affect work relationships between project managers and developers?*

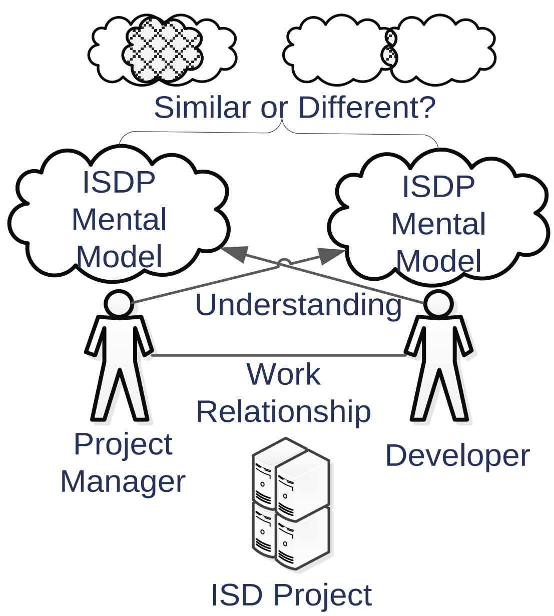


Figure 19 An Illustration of Research Questions

I chose work relationships between project managers and developers as a follow-up study for two reasons. First, since the ISDP mental model construct is newly-developed, the interaction between the dyad instead of the interaction in a collective environment (i.e., project team as a whole) is more suitable for an examination of its potential impact. Second, I contend that the premise of effective work relationships should rest upon ISDP mental models. As suggested in Section 1.2, project managers and developers rely on their ISDP mental models to manage complex, interdependent, and often unpredictable project demands, and rely on their understanding of one another's mental models to anticipate actions. However, the tension can arise from dissimilar knowledge and beliefs; for instance, people who hold customer-oriented and product-oriented beliefs (either project managers or developers) and believe in concepts, such as "iterative and incremental development", "continuous integration", and "promote simplicity" would disagree with their counterparts, who hold the enterprise-oriented belief about spending months on planning out the development process (e.g., "plan-driven development" and "conscious efforts to make project size, cost, and schedule estimation" concept). The task conflict can create a biased understanding of each other and in turn escalates conflict. Despite the fact that similarity in general breeds attraction (Byrne, 1971) and similarity in team mental models facilitate team processes and performance (Mathieu et al., 2000), there is little knowledge about the impact of having similar knowledge and beliefs in ISD project management and if the similarity of mental models helps work relationships. This leads to the first research question: *How does the similarity of ISDP mental models between project managers and developers affect work relationships?*

I also suggest that one's (mis)understanding of co-workers' mental models has particular relevance to work relationships. ISD tasks demands an integration of diverse knowledge, beliefs, and perspectives to solve problems. Extending from research on interpersonal interaction (Allport, 1954) and team cognition (Cannon-Bowers, Salas, & Converse, 1993; Wegner, 1995), it is conceivable that an accurate understanding of co-workers' mental models should be able to help identify expertise and elaborate on one another's knowledge. Besides the task aspect of the work relationships, both project managers and developers would be able to appreciate each other's professional credentials and work contributions. An accurate understanding of co-workers' mental models may help reduce misunderstanding caused by cognitive biases (e.g., stereotypes of project managers). This leads to the second and third research questions: *How does an understanding of others' ISDP mental models affect work relationships?* and *what are the major drivers of an accurate understanding of others' ISDP mental models?*

In the following sections, I will first explore potential conflicting relationships between project managers and project members. Next, the impacts of similar mental models and an accurate understanding will be discussed individually. This is followed by research methodology, results, and discussion.

4.1 Work Relationships in a Nutshell

Work relationships - "patterns of exchanges between two interacting members or partners, whether individuals, groups, or organizations, typically directed at the accomplishment of some common objectives or goals" (Ferris et al., 2009, p. 1379) - are fundamental behaviors in organizations. Literature has indicated that various types of work relationships (e.g., coworker-coworker, supervisor-subordinate, employee-employer) are consequential to an individual's attitudes and behaviors (e.g. employee job satisfaction, commitment, turnover, organizational citizenship behavior and performance), team effectiveness, and organizational functioning and performance(Coyle-Shapiro, Shore, Taylor, & Tetrick, 2004; Dulebohn, Bommer, Liden, Brouer, & Ferris, 2012). Benefits of the positive work relationships come from the content of exchange and reciprocity. Either party receives economic (e.g., information, money, services) or socio-emotional (e.g., love, status) resources which address one's work, social, and esteem needs (Cropanzano & Mitchell, 2005). Once reciprocal interdependence has been developed, relationship development becomes a virtuous cycle with built-in trust and respect. Both parties are willing to provide more support or make commitments beyond an individual's job requirements, according to reciprocity (Gouldner, 1960) and social exchange theory (Blau, 1964).

Work relationship is an overarching concept that encompasses multiple interpersonal constructs. Ferris et al. (2009) identified multiple dimensions of work relationships from different streams of literature[20]. Their literature review represents a predominant focus of work relationships on perception of people, which is a type of attitudes, such as trust, affect, and respect. People form perceptions of their co-workers based on their knowledge, traits, disposition, intention, and social category (Klimoski & Donahue, 2001). The judgment can be formed based on information from other co-workers, observation, and experience during work interaction. For instance, given a coworker's deep knowledge in a task domain, disposition to do good to others, and intention to accomplish project goals, we may show trust, support, and respect to him/her. It also can be formed through heuristics, such as social categorization, without any factual information supporting the judgment. For instance, a co-worker who is in his 50s and is a programmer may not be good at interpersonal skills. Besides the perceptional aspect of work relationships, behaviors transpiring during the exchange process are also components of work relationships, such as the act of support, backup, and helping. Interestingly, despite the emphasis of work relationships on "patterns of exchanges", behaviors are more often considered an outcome of work relationships (see constructs such as citizenship behaviors and coordination behaviors) rather than a process of work relationships. I take both the perceptional and behavioral aspects of work relationships into consideration. However, since the relationship between mental models, understanding, and work

---

[20] There are nine dimensions coming from the literature of leader-member exchange, mentoring, positive connections, social networks, relationship science, and employee-organization relationships. These nine dimensions are trust (positive expectation toward the future), support (the act of upholding, giving faith and confidence to, or otherwise corroborating another person), affect (liking and attraction), respect (positive judgments of past exchanges), loyalty (public backing of one another), accountability (meeting the expectations tied to maintaining high-quality relationships), instrumentality (the relative value an individual perceives from a dyadic relationship), flexibility (The capacity and willingness to be tractable, adjustable, and modifiable).

relationships is less clear, I follow Eisenhardt's (1989) suggestion that identification of constructs may be useful to shape the design but whether the constructs are in a resultant theory should not be predefined. I let the theoretical relationship between them emerge from the data. This research is an exploratory research aiming to build a theory on work relationships between project managers and developers using a mental model perspective.

Before introducing cases, I begin with the potential task interdependence and conflict between project managers and developers. Subsequently, I provide the meta-theoretical background where I present what is known about mental models, understanding, and work relationships.

4.2 Work Relationships between Project Managers and Developers

There are a variety of concerns for work relationships in organizations, such as leader-member exchange, mentoring, social networks with internal and external actors, workplace romance, and employee-organization relationships (Ferris et al., 2009). Previous studies in IS have examined the importance of work relationships between developer-user (Beath & Orlikowski, 1994; McKeen, Guimaraes, & Wetherbe, 1994), developer-developer (Yang & Tang, 2004), project manager-top management (Xia & Lee, 2004), and project member-external agent (e.g., consultants) (Levina & Vaast, 2005). Study 2 focuses on work relationships between project managers and developers because effective work relationships between project managers and developers are essential to the success of a project. Although project managers play the key role of the management of project activities and people (Jiang, Klein, & Chen, 2001), it takes two to tango. Both project managers and developers need to work together not only on regular tasks over the course projects, such as planning, scope definition, and issue tracking, but also on resolving issues arising from customers, top management, contractors, or a team itself, such as requirement volatility and technological uncertainty.

Despite the need of close and effective work relationships, work relationships in a project setting often begin with uncertainties and ambiguities because of a lack of shared knowledge and beliefs among project members. Consistent with our results in study 1, prior research indicates diverse knowledge in the ISD field (Jayaratna, 1994) and various beliefs in project outcomes (Huisman & Iivari, 2006). Some believe a team-centric approach is the best, while others believe one must start with enterprise architecture. Some believe defined procedures expedite learning and ensure control while others see them as rigid and unrealistic to enact changes. While diversity brings a possibility to be creative and adaptive to volatile projects, diversity, without appropriate management, creates fissures between project managers and developers.

Differences between project managers and developers can be further attributed to goals and identity. Studies have shown that project managers and developers espouse different goals due to different roles and responsibilities. The former focuses more on schedule and budget while the latter desires technical excellence and personal development (Mahaney & Lederer, 2003). If one of them does not work toward integrative goals and instead blocks the attainment of the other's goals, the dyad will experience mutual distrust and frustration, thereby deteriorating work relationships.

Identity is about "how individuals define themselves vis-à-vis others and how they are identified by them" (Lührmann & Eberl, 2007). In the work setting, individuals derive their professional identity from the role definitions and their own work groups. Research has pointed out strong professional identity of developers with salient characteristics including valuing information technology (IT) knowledge and skills, keeping pace with the IT changes, and viewing IT as an enabler of organizational changes (Guzman, Stam, & Stanton, 2008; Ramachandran & Rao, 2006). With regard to project managers, although they often juggle different roles and wear different persona (Napier, Keil, & Tan, 2009; Roberts

& Fusfeld, 1997), with the advent of the project management discipline, project managers become to have a strong professional identity that values project management as a complex discipline (Hodgson, 2002).

Identity differences is psychologically salient to create distance between the in-group and out-group as suggested by social identity theory and self-categorization theory (Hogg & Terry, 2000; Turner, 1987). Accordingly, project managers are likely to be seen as out-group members from developers. If both parties cannot find common ground, cooperation, interaction, and relationships will all be challenging (Hogg, 2001). The work relationships are even worse if two professional groups do not appreciate one another's importance. In reality, one party often does not respect the knowledge and capabilities of the other's, which falters meaningful interaction.

Such goal incongruence (Kristof-Brown & Stevens, 2001) and a lack of identity comprehension (Thatcher & Greer, 2008) substantially reduce job satisfaction and jeopardize project outcomes. The following quotes vividly depict incongruent goals and different identities.

> *"Over the course of my career I have dealt with legions of formal "project managers" (folks who are pure project managers lacking any technical background) and I have yet to realize any value in my interactions with any of them, beyond the occasional willingness to record meeting minutes. To date I have found them to be glorified secretaries, whose primary tactic is to latch on to knowledgeable people and not only drain information but actually get them to perform the real tasks of project management, such as scheduling and resource estimation"* (Developer A in Case and Piñeiro (2009))

> *"Whether its IT, Municipal drafting Electrical or whatever, Engineers (regardless of how long they have 'managed' projects) are NOT Project Managers. You frustrate the hell out of me. I've been a Professional Project Manager for years and an Amateur computer geek. The thing that always stuck in my craw is the assumption that just because a person knows an Engineering Discipline that they automatically know how to manage projects. Project Management is a complex discipline and to manage projects well takes a solid educational background in that arena. It is a skill set unto itself. Document Controls, managing Gaant [sic] charts and schedules*

and (especially) managing the 'people' end of things takes a great deal of effort to excel at. But NOOOOOO, Engineers always assume that because they can conceive a project, they MUST be able to manage it, and it always ends up as a grand jitterbug called, 'Crisis Management'" (Project Manager A in Case and Piñeiro (2009))

The consequence of strained relationships between project managers and developers could be far-reaching. The exchange behaviors in a strained relationship, at best, will be purely transactional: project managers give orders to developers about what they need to do and developers merely complete tasks based on their job responsibilities (M. Uhl-Bien & Graen, 1993). Long term reciprocity and social exchanges are less likely to grow. Trust could not develop through the process. Several streams of research have shed light on the impact of strained work relationships.

First, trust is central to relationship development. It is comprised of two components: "the willingness to be vulnerable" (Mayer, Davis, & Schoorman, 1995) and "positive expectations about other" (Lewick & Bunker, 1996). Strained work relationship often means that project managers and developers do not hold positive expectation about others. When facing ambiguous course of actions in projects, they cannot rely on another part and are unwilling to take risk during interaction. The cost of control and monitoring surges and friction between parties becomes unavoidable. Several meta-analyses in general management have shown that a lack of trust reduces job satisfaction and performance (Colquitt, Scott, & LePine, 2007; Dirks & Ferrin, 2002) and even produces counterproductive behaviors (Colquitt et al., 2007). In ISD teams, while not directly examining trust between project managers and developers, research found trust is beneficial to knowledge exchange (Joshi, Sarker, & Sarker, 2007) and results in less monitoring behaviors between each other (Serva, Fuller, & Mayer, 2005).

Leadership literature suggests similar findings. Leader-member exchange (LMX) theory suggests that leaders form different types of exchange relationships with their subordinates and cause different subordinate behaviors and performance (Graen & Uhl-Bien, 1995). The quality LMX relationship is contingent upon the exchange of valued resources and emotional support. The low quality of exchange has substantial negative impacts on individual attitudes and performance, such as commitment, job satisfaction, and job performance (Dulebohn et al., 2012). Typically project managers are not direct supervisors of project members and have limited institutional power to make critical decisions that affect project members, such as pay raises, promotions and demotions, and training opportunities. However, project managers typically play a key role within project teams and influence the environment in which the project team is embedded for project success; for instance Ozer (2008) found that when developers are provided with freedom to do their jobs and the quality of LMX high leads to better job satisfaction. In the context of strained work relationships, project managers may not be able to move the project smoothly, let alone motivate project members to transcend their own self-interest in pursuit of group goals.

Additionally, research about co-worker exchange relationships can also inform on the potential consequences of ineffective work relationships. There is close task-interdependence between project managers and developers, particularly in many small to medium ISD projects where project managers wear different hats at the same time (e.g., project manager as a business analyst, Gilbert, 2004). When relationships turn sour, the dire consequences are not limited to individual counter-productive work behaviors and lower job satisfaction but also include immediate task outcomes (Chiaburu & Harrison, 2008).

To mitigate risks caused by ineffective work relationships, I will first briefly touch upon the antecedents and potential interventions of forming work relationships, and then explicate the potential of similar mental models and accurate understanding.

4.3 Mental Models and Work Relationship Development

To maintain work relationships, various factors need to be considered. Different theoretical perspectives have been adopted in the literature. Some researchers believe that similarities of individual characteristics such as demographic attributes (Green, Anderson, & Shivers, 1996), personality (Bauer & Green, 1996), cognitive styles (Allinson, Armstrong, & Hayes, 2001), and individual values (Ashkanasy & O'Connor, 1997), increase liking and improve work relationships. Others pay attention to behavioral patterns that are consistently related to relationship improvement, such as communication and interaction (J. Lee, Graen, & Graen, 2005) and the content as well as form of social exchange behaviors (Cropanzano & Mitchell, 2005). Still others propose a contingency theory and emphasize the role of contextual factors in work relationships, such as task autonomy (Ozer, 2008), organizational climate (Dienesch & Liden, 1986), and anchoring events (Ballinger & Rockmann, 2010). To bridge multiple perspectives from individual characteristics, communications and behaviors, to contextual factors, I adopt a cognitive perspective, using the ISDP mental model construct, because cognition guides individuals' actions and decisions. In the following sections, I will first recap the ISDP mental model construct and then I will explore how similarities and accurate understanding may affect works relationships.

4.3.1 Recap: The ISDP Mental Model

A project member's ISDP mental model is defined as the knowledge and belief structure that help him/her to understand, conduct, and manage ISD projects. Knowledge structure is a semantic understanding of how different concepts are meaningfully related (Sparrow, 1998). For instance, the concept about releasing software frequently to customers and the concept about involving users are related because both aim to gather feedback and deliver business values. Knowledge structure allows individuals to sort out messy information and make sense of the environment. Beliefs rest on a stored body of structured knowledge and indicate relationships between concepts, objects, people, etc. (Frijda, Manstead, & Bem, 2000). ISD professionals have broad knowledge about managing ISD projects and believe that certain concepts can achieve desirable outcomes. Such evaluative beliefs direct cognitive processing in the acquisition of new knowledge (Nystrom & Starbuck, 1984).

The results from study 1 suggest 40 fundamental ISDP concepts (see Appendix H) and four evaluative beliefs: team, customer, product, and enterprise. The ISDP mental model acts as a frame of reference to guide project members in their behaviors and anticipation of other's behaviors, such as to what extent project members should interact with users, development procedures should be formalized, and project members should participate in planning and estimation. Unlike other types of IS-relevant mental models that focus on independent tasks (e.g., technology knowledge for programming), the ISDP mental model underlies every project member's project experience and continues to permeate project life, including communication, coordination, and major decisions. The collision between different ISDP mental models becomes apparent along with the burgeoning number of methodologies, global collaboration, and relatively frequent merger and acquisition. In order to identify real differences between project managers and developers, a systematic analysis of individual mental model is necessary (using the assessment tool I developed in Study 1).

4.3.2 Similar ISDP Mental Models and Work Relationships

Similarities (and dissimilarities) between mental models of people in ISD teams have the potential to influence interpersonal interaction in several ways. From an information processing perspective, similar mental models facilitate meaningful communication via common language and shared knowledge (Krauss & Fussell, 1990; G. Lee & Xia, 2010; Preston & Karahanna, 2009) and frame the problems in similar ways (Cronin & Weingart, 2007). Similar mental models in terms of work values, norms, philosophy, problem-solving approaches, and work experience facilitate knowledge transfer between ERP consultants and clients (Ko, Kirsch, & King, 2005). Moreover, similar mental models regarding development processes, problem domains, and project team contexts help distributed ISD teams coordinate (Espinosa et al., 2007). On the contrary, dissimilar mental models indicate different ways of implementing projects and different priority on project outcomes. Over time, if the differences are not reconciled, either through the shift of mental models, negotiation, or perspective taking, frustration can build, turning task and process conflict into relationship conflict. Negative emotions are not helpful for task performance. Research has found that software project members who have diverse backgrounds are less willing to help or be loyal to other project members especially when they work in highly independent tasks and lack shared goals and outcomes (or vice versa) (G. S. Van Der Vegt, Van De Vliert, & Oosterhof, 2003).

From the social cognition perspective, similar mental models imply the existence of liking, as suggested by similarity-attraction paradigm (Byrne, 1971). Similarities increase a sense of familiarity and safety during the interaction (Klohnen & Luo, 2003) and satisfy people's fundamental need for consensual validation of their perspectives (Turner, Brown, & Tajfel, 1979). In IS teams, team members who have similar knowledge and beliefs in works are emotionally attached to their teams (Yu, Hao, Dong, & Khalifa, 2013) and are more willing to help other team members (G. S. Van Der Vegt et al., 2003). Team members therefore are more likely to have effective work relationships. Likewise, social identity theory suggests that people categorize others who are similar as in-group members and who are different as out-group members (Allen & Wilder, 1975). Interpersonal differences increases psychological distance and lead to detachment and interpersonal conflict (Mary Uhl-Bien, 2006), thereby negatively affecting work relationships. There has been considerable conflict between ISD professionals and users due to the intergroup bias (Chang, Chu, Chi, & Lo, 2010). While the intergroup bias is not examined between project managers and developers, the stories of the conflict have been circulated as can be seen in the following instance.

> *"I think it's important for people to read this and realize that the PM is the reason that programming wreck it by having no concept of reality. I don't much care if the PM doesn't understand exactly HOW to implement things, but it's been my experience (most recently as a*

lead programmer at a small outfit) that the PM will make decisions based on the immediate costs, rather than any solid scientific research." (Developer C in Case and Piñeiro (2009))

Arguably, the intergroup biases may cause similar problems between project managers and developers.

The social capital perspective also lends insight into the relationships between similar mental models and work relationships. Social capital that is embedded in effective work relationships is a valuable asset to provide access to knowledge and it also motivates individuals to exchange knowledge (e.g., Levin & Cross, 2004). In order to build and maintain effective, repeated, and enduring work exchanges, similar mental models are needed, among other structural (e.g., tie strength) and relational (e.g., trust) aspects of social capital (Nahapiet & Ghoshal, 1998). Along with aforementioned cognitive processing and socio-cognitive needs, the social capital perspective additionally emphasizes the importance of similar mental models in constructing shared meaning within work environment.

Prior literature primarily investigates mental models at the generic level (Mohammed et al., 2010) and touches upon perceived similarities in terms of project goals, task domain, procedures, technologies, communication channels (e.g., He et al., 2007; Yang et al., 2008). Particular knowledge and beliefs toward the management of ISD projects are rarely examined, let alone topics regarding how knowledge and beliefs are correlated in ISD professionals' minds. This study looks into actual similarities in terms of knowledge and beliefs between project managers and developers (see the methodology section for details). The findings can potentially help academics and practitioners learn what problems would be caused by specific dissimilar knowledge and beliefs between project managers and developers.

4.3.3 Accurate Understanding of Mental Models

Despite the impact of the degree of similarity of mental models, I believe that work relationships can be sustained and improved even when interacting partners do not have similar mental models. According to the theory of complementary needs (Wagner, 1975; Winch, Ktsanes, & Ktsanes, 1954), people should be attracted to those who have mental models that are complementary to their own. Complementary mental models usually represent dissimilar knowledge between the dyad but their knowledge combined together can make up for one another's shortage and serve the dyad's purpose. For example, project managers may not have knowledge of defect prevention and code reuse, but is a strong believer of collaboration, motivating performance, and training. She/he may seek help from project members who is tech-savvy. At the same time, project members need resources and support from project managers. A project manager who is willing and able to acquire resources and offer support fulfills project members' needs. While instrumentality dominates the initial social exchange, trust, support, and respect could grow over time, thereby bolstering the relationship.

The theory of complementary needs implies that individuals should have an accurate understanding of other people so that they are able to identify who have knowledge and find ways to collaborate with those people who have different mindsets. An accurate understanding of other's knowledge and beliefs is needed to ensure that the dissimilar mental models do not result in negativity. The value of accurate understanding is also suggested by the Contact Hypothesis (Allport, 1954): an enhanced understanding of interacting partners, particularly among the dyad that are experiencing conflict or potentially have conflicts, is an essential mechanism for improving relationships because it mitigates stereotyping, prejudice, and discrimination.

The concept of understanding permeates various disciplines, including organizational behaviors, psychology, management information systems, and strategy. The theory of transactive memory systems (TMS) involves a set of individual memory systems in which communications between dyads facilitate the development of collective memory systems (Wegner, 1995). It is suggested that an understanding of other people's knowledge helps identify expertise and retrieve knowledge (Faraj & Sproull, 2000). Therefore, project members can focus on their own areas and capitalize on one another's expertise. The process of cognitive division of labor stimulates team learning and communicates knowledge among team members. Clearly, TMS indicates that in the organizational context the task knowledge and skills held by each team member should be accurately understood.

Another similar construct from shared cognition research is team mental models (TMM). TMM is the organized mental representation of the key elements within a team's relevant environment that are shared across team members (Klimoski & Mohammed, 1994). It is crucial to team effectiveness when a team is equipped with accurate and similar mental models in terms of taskwork and teamwork. Shared taskwork mental models (e.g., knowledge about technology, equipment, procedures, strategies) ensure common task representations and task executions while shared teamwork mental models (e.g., information flows, communication, and awareness of member responsibilities and characteristics) facilitate interaction to complete the tasks (Mathieu et al., 2000). The term "shared" implies two elements of TMM. One is the degree of similarity, which suggests that the certain content and organization of knowledge about the tasks and teams should overlap. Depending on the context, teams require different degrees of overlapping. An example provided by Cooke et al.(2000) is a surgical team where a surgeon and a nurse only need similar task-specific knowledge to a minimal degree but need similar task-related knowledge to a considerable degree. Another aspect is the awareness of the similarity, which suggests benefits of conscious awareness of what they share (i.e., what they know and what they know about others) (Cannon-Bowers, 2007). The awareness, which is largely overlooked in the existing TMM literature, involves the process of understanding. Team members rely on their understanding of others' task knowledge to attribute others' behaviors (Bazerman, Curhan, Moore, & Valley, 2000; Rentsch & Zelno, 2005) and to integrate each other's ideas and approaches to solve task issues (Cronin, Bezrukova, Weingart, & Tinsley, 2011).Besides, similar to TMS, an understanding of skills and abilities helps understand where expertise is located and who needs knowledge. Furthermore, an understanding of team members' skills and abilities helps elaboration and utilization of task information (Hsu et al., 2011).

To comprehensively explore the meaning of understanding, I draw on the cross-understanding construct proposed by Huber and Lewis (2010). Cross-understanding is an accurate understanding of one another's mental models in terms of factual knowledge, beliefs, sensitivities, and preferences. I will next explain how different aspects of understanding in this definition are associated with an understanding of other project member's ISDP mental models.

Factual knowledge is about declarative and procedural knowledge stored in knowledge structures. An understanding of others' factual knowledge helps avoid using jargon and unique knowledge that are likely to be incomprehensible for people who do not have such knowledge. In ISD projects, developers who talk primarily on the importance of technical knowledge, such as refactoring and design patterns, may have difficulty in communicating with project managers whose expertise lies in project and business areas. In spite of dissimilar knowledge structures, they are likely to gain benefits from an accurate understanding of factual knowledge. First, they could tailor their language to the interacting partner or they could work together by learning shared vocabularies. This builds common ground that facilitates interpersonal communication. Second, an accurate understanding of factual knowledge helps identify expertise and recognize what could be exchanged. Project managers are able to match developers and tasks, recognize developers' contribution, and integrate expertise. On the other hand, developers can seek resources and support from project managers and offer suggestions on the areas that project managers are not familiar with. In this study, 40 ISD concepts represent factual knowledge.

Belief in Huber and Lewis's (2010) work is referred to as the cause-effect belief, which is the credible relationship between concepts in mental models. Dissimilar beliefs about means-ends relationships facilitate adaptation in turbulent project environment by stimulating debate about appropriate ways to achieve outcomes (Liang, Liu, Lin, & Lin, 2007). However, as often seen in the interaction between people who have different beliefs, people are more likely to scrutinize and challenge each other's ideas (Jehn et al., 1999). The confrontation could be misinterpreted as personal criticism, leading to relationship conflict. An accurate understanding of one another's beliefs means the interacting partner's behaviors and decisions are expected. This helps to avoid misinterpretation. For example, a project manager believes that a rigorous control of development processes results in better project outcomes in terms of quality and schedule, developers who do not know the project manager's belief may attribute such behavioral control to a lack of trust, which ruins work relationships. In this study, the relationships between concepts indicate the cause-effect belief.

Preference is related to the desirable states and goals that individuals prefer, which is also called values or evaluative beliefs (Huber & Lewis, 2010). Incompatible goals have been the source of interpersonal conflict (Barki & Hartwick, 2001). Although project managers and developers may give importance to different goals, they are not in a zero-sum game. The crux of the matter for both is project success. Both parties must frequently negotiate to get what they want. Positive work relationships occur when incompatible goals are settled. This perspective coincides with conflict resolution literature. Individuals' perceptions of goal interdependence affect how they work with other people (Deutsch, 1973). In order to promote the interaction effectiveness, it is critical to shift independent goals and competitive goals to cooperative goals. Since ISD is a social and interactive process, people have to show concern about others to be able to work together (Pruitt & Robin, 1986). An accurate understanding of each other's goals helps one avoid obstructing goals of others and find integrative goals. In this study, four evaluative beliefs represent preferences.

Sensitivity is about whether a particular issue should be considered in a problem representation (Huber & Lewis, 2010). For instance, project managers are sensitive to risks, budget, and schedule issues;

programmers are sensitive to new technologies; senior managers are sensitive to organizational politics. Sensitivity to issues can be driven by preferences or roles. We argue that sensitivity in ISDP mental models can be observed through knowledge structures. That is, issues individuals care about will be salient concepts in mental models. However, given the obscure role of sensitivity in ISDP mental models, sensitivity is not included in this study.

To clarify terms and tie them back to the ISDP mental models of study 1, I define the knowledge structure as a combination of factual knowledge (concepts) and cause-effect beliefs (the relationships between concepts) and the belief structure as preferences. For cross-understanding investigated in this study, I examined the degree to which knowledge and belief structures are understood by project managers and developers.

It is not yet clear when and how an accurate understanding is developed. Similar mental models do not necessarily lead to cross-understanding. People could easily make an inaccurate evaluation of other's mental models by either assuming that others have similar mental models (false consensus bias) or assuming that their knowledge and belief are different from others (pluralistic arrogance bias) (Randolph-Seng & Norris, 2011). Despite this, individuals' knowledge and belief structures still play a critical role for an accurate understanding in the long run. Without similar or relevant knowledge or beliefs, it is difficult to make sense of others' thinking and behaviors (Bower & Hilgard, 1981). For example, individuals who know little about IT governance are not able to recognize their colleague's expertise in IT governance frameworks, let alone why their colleague believes in the power of IT governance on project management. Research has also shown that CIO and other top management team members who have one another's domain knowledge to certain extent tend to demonstrate an accurate understanding of each other and the task (C. P. Armstrong & Sambamurthy, 1999; Bassellier & Benbasat, 2004; Reich & Benbasat, 2000). As stated in Transactive Memory Systems (TMS) and the theory of complementary needs, people gain benefits in collaboration when they have dissimilar mental models. It appears that there should be certain similarities in terms of knowledge and belief structure in order to reap the above benefits. However, little is yet known about how mental model similarities and understanding are related to each other.

4.4 Research Methodology

4.4.1 Choice of Methodology

The purpose of this study is to explore how cognitive factors (i.e., mental models and cross-understanding) contribute to the development of work relationships. Give its complexity and dynamic nature, a study on cognitively anchored work relationships does not readily lend itself to experimental or survey research. Relevant variables for work relationships are too many to control. Additionally, operationalization and the measure of cross-understanding are far from mature. These constraints make experimental or survey research less feasible. Therefore, I chose to pursue our investigation inductively, relying on a qualitative case study approach. Case study is a well-accepted approach to study the “how” or “why” some social phenomena work especially when the phenomenon is poorly understood (Yin, 2009). Through describing and interpreting interaction between project managers and developers, this study will be able to well address the dynamic relationships between cognition and work relationship development. The appropriateness of the qualitative case study research can also be attributed to the fact that individual cognition (either mental models or cross-understanding) and work relationships do not exist in vacuum but are situated in social, political, and economic contexts. Separation of cognition and work relationships from the context would invalidate the findings. The qualitative case study research provides rich descriptions of the context and the participants’ minds. Lastly, I posit that the qualitative case study research suits our research questions since formal and testable propositions are less likely to be generated from current literature. Motivation for exploring mental models and cross-understanding on work relationships is driven by grand theories, such as similarity-attraction paradigm and the theory of complementary needs, and constructs, such as team mental models and transactive memory systems. The consequences of the interaction of mental models and cross-understanding are less certain due to limited theoretical support and inconclusive evidence. The richness of data in the qualitative case study research should be able to provide more meaningful theoretical explanations (T. W. Lee, 1998).

4.4.2 Case Recruitment

The unit of the study is work relationships between project managers and developers. Developers could be business/system analysts, business/system architects, programmers, or testers. I focused on the projects manager-developer dyad in an ISD team in which the dyad is highly task interdependent. High task interdependence represents the existence of intensive work-relevant exchanges in the dyad and allows researchers to observe changes in cross-understanding and work relationships. Additionally, I targeted ISD projects that involve complicated tasks. High task complexity for ISD projects, including not only IT complexity but also organizational complexity (Xia & Lee, 2003), requires new or novel solutions rather than following standard procedures. Project managers and developers have to attend to their own knowledge structures and belief structures and rely on each other to solve non-routine problems. As a result, mental models become critical for the work interaction. The dyads that meet the above criteria provide rich data for theorizing and conducting a detailed analysis of the dynamics of work relationships.

Through personal contacts and recruitment notices posted on online professional forums (e.g., Agile Ottawa Linkedin Group), three organizations in Canada expressed their interests in this research. The first organization was a software design firm (SWCo), providing software engineering services and user experience design on desktop, web and mobile platforms. The second organization was a technology group in a large consulting firm (ConsultCo), offering implementation of ERP systems in the higher education sector. The third organization was an IT department in an insurance company that is responsible for internal and external enterprise applications (FinanceCo). All three organizations have a reputation as a leading-edge firm in terms of their maturity in the development of information systems.

The contact of each organization sent an invitation letter to potential participants (based on the aforementioned case selection criteria). Around 20 pairs expressed their interest in participation. In the end, 3 pairs in SWCo, 3 pairs in ConsultCo, and 2 pairs in FinanceCo completed the participation. One pair in FinanceCo withdrew after the preliminary survey due to their workload. The results of one pair in ConsultCo were removed because the pair did not meet the case selection criteria (due to the size of the project, a developer did not directly report to a project manager and had limited interaction and understanding of the project manager). This resulted in 6 usable cases. Table 6 provides a summary of the sample.

Table 6 Background of Participants and Dyads

Case ID	Role	Experience in IS (years)	Expertise	The number of years/projects working together
P1	Project Manager (P1-PM)	10+	business strategy, business analysis, project management	2 years and 2 projects (continued work relationships)
	Developer (P1-DEV)	2+	programming, project management, team development	
P2	Project Manager (P2-PM)	6+	project management, team development	1.5 year and 3 projects (continued work relationships)
	Developer (P2-DEV)	10+	System analysis, system architecture, programming	
P3	Project Manager (P3-PM)	14	project management, business analysis, team development	1.5 years and 2 projects (not working together now)
	Developer (P3-DEV)	15	programming, system architecture, testing	
P5[21]	Project Manager (P5-PM)	20	business analysis, system analysis, system architecture	1.5 year and 1 project (continued work relationships)
	Developer (P5-DEV)	3+	system analysis, programming, and testing	
P6	Project Manager (P6-PM)	12	system analysis, system architecture, programming, testing	1.5 year and 1 project (continued work
	Developer	9	programming, system architecture, business	

[21] The case was removed for case comparison but stays in this table because the data from the case has been used to support arguments in study 1 and the discussion of limitations in Section 4.6.4.

	(P6-DEV)		architecture, system analysis, testing	relationships)
P7	Project Manager (P7-PM)	9	project management, system analysis, system architecture, programming, testing	2 years and 2 projects (continued work relationships)
	Developer (P7-DEV)	2	business analysis, testing, programming	
P10	Project Manager (P10-PM)	25	project management, system analysis, system architecture	1.5 year and 1 project (not working together now)
	Developer (P10-DEV)	14	business analysis, testing, programming	

4.4.3 Case Categorization

The research design is multiple-case with 2 (mental models: similar vs. dissimilar) X 2 (cross-understanding: high vs. low) design. It was hoped to have two cases in each condition. The cases in different conditions would allow for comparison and contrast of the results (a theoretical replication) while the two cases in the same condition would serve to predict similar results (a literal replication) (Yin, 2009).

Following a theoretical sampling strategy, I categorized cases that vary in the similarity of mental models and cross-understanding. The categorization was based on a survey and the follow-up interviews. In terms of similarity of mental models, I adopted the assessment procedures developed in study 1. Participants were asked to choose 10 important concepts from a list of 40 ISDP concepts (see Appendix J for a snapshot) and then indicate the relationships among the concepts on a 9-point scale from 1 to 9, where 1 denotes unrelated and 9 denotes highly related (e.g., how related "project monitoring and tracking" is to "project transparency") (see Appendix J for a snapshot). The similarity index of ISDP mental models between project managers and developers was computed using the revised distance ratio (Markoczy & Goldberg, 1995) (see Appendix L for the formula and an example). The value of the index is between 0 and 1. A higher value suggests a higher similarity in terms of concepts and the relatedness between concepts.

In our samples (n = 6), the index varies between 0.14 and 0.64 (M = .36, SD = .18) (See Table 7). An arbitrary cut-off was used. The case with the similarity index below 0.25 was considered the dissimilar ISDP mental model condition. For the cases near the cut-off boundary (P2 and P6), I further examined their belief structures using the profile deviation approach (i.e., a comparison of four ideal belief structures and individual mental models, as discussed in Section 3.5.2) (See Table 7). The dissimilar knowledge with similar dominant beliefs suggests that the dyad may hold different concepts and weigh the relatedness between the similar concepts differently but the ultimate goal is similar. They are more likely to accept one another's ideas. Accordingly, P2 is categorized as the similar ISDP mental model condition while P6 is categorized as the dissimilar ISDP mental model condition.

Table 7 Similarity of ISDP Mental Models and Dominant Beliefs

Case ID	Similarity Index	Dominant beliefs[22]
P1	0.48	P1-PM: Customer (.44); P1-DEV: Customer (.44)
P2	0.26	P2-PM: Team (.41); P2-DEV: Team (.35)
P3	0.14	P3-PM: Customer (.38); P3-DEV: Team (.41)
P6	0.3	P6-PM: Team (.43); P6-DEV: Enterprise (.53)
P7	0.36	P7-PM: Team (.54); P7-DEV: Customer (.38)
P10	0.64	P10-PM: Customer (.39); P10-DEV: Customer (.51)

Cases classification of high/low cross-understanding is on the basis of the comparison between individual's ISDP mental models and the interview data from his/her co-worker[23]. Forty concepts and four types of beliefs were used in the coding scheme. Below I describe the procedures of classification.

---

[22] The value in the parentheses represents the similarity index between individual beliefs and four ideal beliefs. The value of the index is between 0 and 1. A higher value suggests higher similarity.

[23] To identify cases with high/low cross-understanding before interviews, I also developed measures to assess one's perceived understanding of the other's knowledge and beliefs. The items for an understanding of knowledge are derived and revised from 40 ISDP concepts of study 1 and the items for an understanding of beliefs were revised from Quinn and Rohrbaugh's (1983) organizational value items (see Appendix M). Both measures were pretested with doctoral students in the MIS area to confirm its content validity and discriminant validity (using a card sorting technique). I did not follow a strict scale development process (Lewis, Templeton, & Byrd, 2005) for two reasons. First, the measures only serve as initial screening for case selection. The validity issue of perceptual measures only increases efforts to conduct more interviews. That is, if I anticipate conducting interviews with a high cross-understanding pair but after the interviews and data analyses, I realize that the pair, in fact, has low cross-understanding, I need to look for another high cross-understanding pair. Second, even with the rigor procedure, the perceptual measure may not accurately capture one's accurate understanding because people may misjudge others' knowledge and beliefs (Huber & Lewis, 2010).

Regarding an accurate understanding of knowledge structures, I first developed the pool of concepts each participant has by combining the data from the survey (i.e., assessment of ISDP mental models) and interview (interviewees described their knowledge and beliefs about the management of ISD projects, see Section 4.4.4). Participants were restricted to choose 10 most important concepts in the survey. Some important concepts, which were not picked by participants in the survey, may not be recorded. Hence, I coded the interview data and added concepts into the pool. On average, two more concepts were added to each person's ISDP mental models (minimum: 1; maximum: 4). Second, I examined the accuracy of understanding by comparing participants' interview responses (interviewees described their co-worker's knowledge about the management of ISD projects, see Section 4.4.4) with their co-worker's pool of concepts. The accurate understanding is confirmed when the results match. For instance, P1-PM knows and applies the concept of "reflect on improvement at regular intervals" and P1-DEV made the following comments:

> *"So he [P1-PM] saw a process that worked, and he's learned that from experience. He's learned that from the theory, and he enforced it thoroughly. So retrospect, he never swayed. I've seen… I've worked with project managers where we would have a retrospect and then perhaps we'd have another retrospect, and then that's it for the rest of the project. Like, somehow the meetings got misplaced. So he never missed that, improvement, learning and development."* (P1-DEV)

P1-DEV was considered to have an accurate understanding of P1-PM's "reflect on improvement at regular intervals" concept.

Regarding an accurate understanding of beliefs, participants were asked to describe outcomes that their co-worker values. Their accounts were compared with the survey results regarding beliefs. For instance, P3-DEV described his understanding of what P3-PM values.

> *"P3-PM takes it [customer satisfaction] seriously…She puts a lot of effort into making sure the customer is being listened to and that she's helping to interpret and understand their requirements and make sure that they're getting what they're asking for…She would want to see*

the project be embraced by the intended audience. I think she would want to see that the team as a whole genuinely enjoys working together on the project" (P3-DEV)

These accounts were consistent with P3-PM's beliefs (based on the assessment of ISDP mental models).

It should be noted that the absolute degree of understanding (e.g., a co-worker needs to show his/her understanding of at least 80% of ISDP mental models) was not the major criterion of this study. The decision was made because the limited duration of the interview constrained participants from talking about all important knowledge and beliefs their co-worker may hold. However, for pairs who were considered high cross-understanding pairs, they accurately described at least 50% of concepts their partners possess. They also covered most important outcomes that their co-workers believe.

In addition to the accurate understanding of knowledge and beliefs, I assessed perceived accurate understanding by asking participants whether they think they know their partners well professionally. When the pairs had an accurate understanding of knowledge and beliefs along with high perceived understanding, they were classified into the high cross-understanding condition. Table 8 represents the case classification.

Table 8 Research Design

	High cross-understanding	Low cross-understanding
Similar mental models	P1 and P7	P2 and P10
Dissimilar mental models	P3 and P6	

There is no case in the dissimilar mental models and low cross-understanding condition. A couple of reasons hindered the data collection. First, the topic of discussion is about work relationships. It has been challenging to acquire consent to participate from the pairs who have quite different knowledge and beliefs and do not know much about one another because they are likely to have less effective work relationships. As readers may find in the six cases I recruited, all of them have effective work relationships to some extent. Second, according to the attraction-selection-attrition framework (Schneider, 1987), people flock to the same organization because they hold similar knowledge and beliefs. The possibility of obtaining people who have dissimilar mental models is lower. Despite a lack of extreme cases, according to the data analysis, the six cases have substantial variance in terms of mental models, understanding, and work relationships, which provides evidence to answer research questions and provides theoretical saturation (Eisenhardt, 1989).

4.4.4 Data Collection Procedures

A mix of qualitative and quantitative data collection was used in this study in order to triangulate results (Creswell, 2008). Data collection took place at three phases. At Phase 1, I invited prospective participants to fill out an online survey. The questionnaire includes the scales of ISDP mental models developed in study 1 (Appendix J), perceived understanding of the co-worker (Appendix M), prior work experience with the co-worker (Appendix N), and task interdependence (Appendix N). The objective is to select cases that fit the criteria described in Section 4.4.2. I analyzed data (See Section 4.4.3), picked appropriate cases, and contacted participants for interviews at Phase 2.

At Phase 2 (around 3 - 4 weeks after Phase 1), semi-structured interviews were conducted over a four-month period. Project managers and developers were interviewed separately and on average the interview lasted for 62 minutes (SD = 11.06). With the participant's consent, the interviews were all audio-recorded. The data collection process was centered on what project managers and developers understood about one another and their working relationships. I first asked the interviewees' software development experience, expertise, preferred practices, and beliefs. Then, the interviewees were asked about their understanding of the dyadic partner's knowledge and beliefs in the management of ISD. Interviewees were also inquired about how the similarity and their understanding of one another influence the development of work relationship (Research Question 1 and 2). Lastly, interviewees were asked to describe their work relationships with the dyadic partners in general and provide examples of effective working experience and conflicts in particular. The detailed interview protocol is presented in Appendix P. After the interview, I showed participants their mental models (from Time 0) (see an example in Appendix O) and asked them to confirm the accuracy of mental models.

Given the exploratory nature of this study, I started with a case that has high cross-understanding (P3) to explore influences of cross-understanding on work relationships. The preliminary case provided an opportunity to formulate a set of new theory-based explanations and revise sampling criteria. After preliminary data analysis, the results provided initial support for the role of understanding in the development of work relationships. I proceeded with the remaining cases.

4.4.5 Data Analysis

All interviews were transcribed, coded, and analyzed using MAXQDA version 11, a commercially available software package specifically designed for qualitative data analysis. To address my overarching research question - *How does the interplay of ISDP mental models affect work relationships between project managers and developers?*, I focused on four sets of analyses: (1) confirming the accuracy of ISDP mental models, (2) assessing the accuracy of understanding between project managers, (3) investigating what constitutes the work relationships and how similarity and understanding affect the work relationships, and (4) summarizing factors that may affect accurate understanding.

The first two sets aim to set up a theory development for my research questions by validating the assessment procedures and classifying cases into different conditions, as I have described in Section 4.4.3. The third and fourth set is designed to develop a theory where the analyses followed an iterative process and standard methods described for inductive research (Miles, Huberman, & Saldaña, 2013). Following the guidelines suggested by Miles et al. (2013), I started with a list of master codes that is comprised of "understanding of ISDP mental models" and "work relationships". As the analysis proceeded, I searched for general attributes of the master codes and then compared and contrasted the master codes with different attributes. I continued to revise, add, and discard codes and code sets. The qualitative data analysis software was used to track the content of recurring themes and codes. I was careful to ensure that the particular themes and codes identified were not based on instances related by a non-representative informant (Miles et al., 2013).

Several themes consistently emerged during the discussion about work relationships and understanding. During the first round of analysis, various aspects of work relationships became clear, which can be classified into attitudes (e.g., liking, professional respect, and loyalty) and behaviors (e.g., information exchange and helping behaviors). After several rounds of analysis, it became apparent that trustworthiness incorporates various attitudes and coordination encompasses work-related behaviors. Meanwhile, I discovered that participants discussed a variety of understanding beyond an understanding of ISDP mental models, such as an understanding of personality, an understanding of communication styles, and so on. Among them, the three most salient types are an understanding of expertise, an understanding of needs and constraints, and an understanding of social aspect. The coding criteria presented in Appendix Q reflect the final coding scheme.

4.5 Results

As shown in Figure 20, I have organized findings into a framework that represents the ways in which understanding affects work relationships and three antecedents of understanding. It should be noted that this framework is inductively generated from fieldwork, that is, a theory emerges from my observations and interviews.

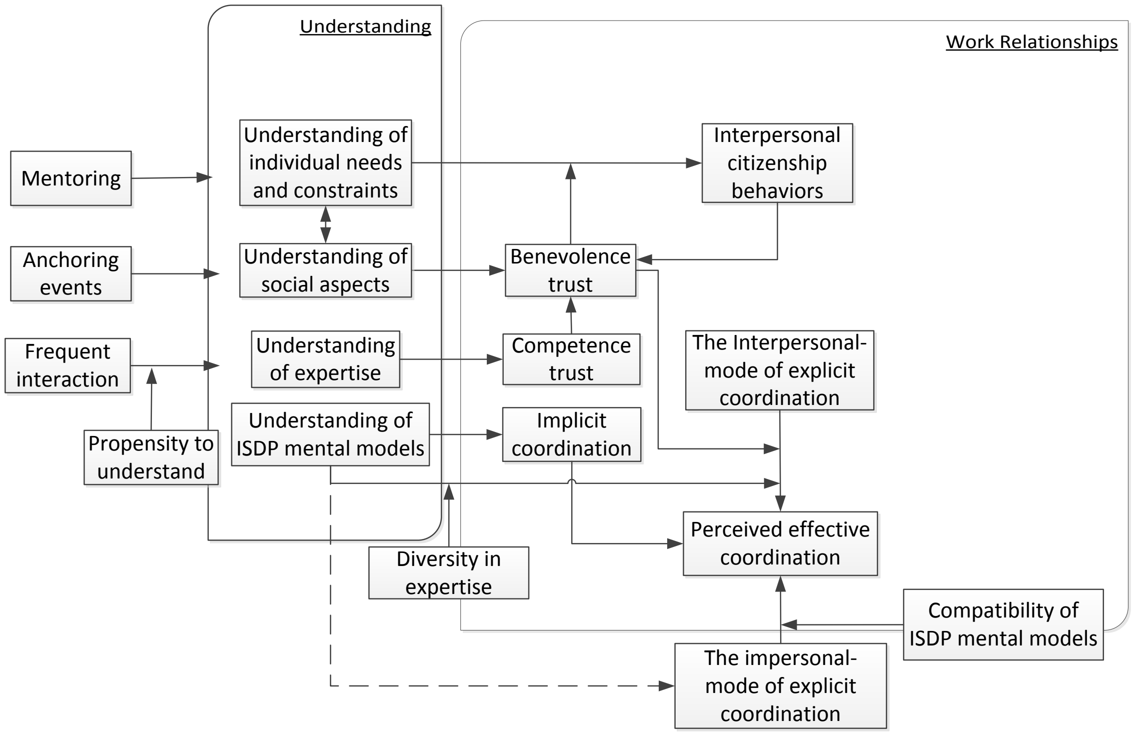


Figure 20 The Theoretical Model for Work Relationships in ISD Projects[24]

I chose to present data by concepts rather than by research questions and cases because there are several new concepts unexplained previously emerging from the data. I will focus on concepts and embed the discussion of research questions and cases within each concept. I deviate from the traditional sequence followed in an input-process-output framework and start with outcomes: coordination, trustworthiness, and citizenship behaviors, in which I explain how three major dimensions of work relationships are influenced by similarity of mental models (specifically why it is not shown in the theoretical framework) and an understanding of mental models. Readers may note that some new concepts are showed in the theoretical framework, such as different aspects of understanding. I will describe how they are related to

[24] Dash line indicates a partial support, requiring future research.

work relationships. Lastly, I address the third research question by discussing three antecedents for an accurate understanding. Table 9 summarizes the results for each case under three case conditions.

Table 9 Result Summary - Dimensions of Work Relationships

		High understanding and High similarity		High understanding and Low similarity		Low understanding and High similarity	
		P1	P7	P3	P6	P2	P10
Implicit coordination	Unsolicited project-relevant information	Yes	Yes	Yes	Yes	No	No
	Adaptation to others' ISDP knowledge and beliefs	Yes	Yes	Yes	Yes	No	No
	Task assignments	Yes	Yes	Yes	No	No	No
Explicit coordination (interpersonal mode)	Information exchange	Yes	Yes	Yes	Yes	Yes	Yes
	Feedback seeking and giving	Yes	Yes	No	Yes	Yes	No
	Knowledge integration	Yes	Yes	Yes	No	No	No
Antecedents	Frequent interaction	Yes	Yes	Yes	Yes	Yes	No
	Mentorship	Yes	Yes	No	No	No	No
	Anchoring events	Yes	Yes	Yes	No	No	No
Compatibility between mental models and the impersonal mode of explicit coordination		Yes	Yes	Yes	Yes	Yes	Yes
Interpersonal citizenship behaviors		Yes	Yes	Yes	Yes	Yes	Yes
Benevolence trust		Very high	Very high	Very high	High	High	High
Competence trust		High	Very high	High	High	Moderate to high	Very high
Perceived work relationships		Above average	Above average	Above average	Above average	Average	Average

4.5.1 Coordination

Coordination refers to the management of interdependence among tasks (Malone & Crowston, 1994) and the management of interdependence between people who carry out tasks (Gittell, 2001). Coordination is regarded as a prominent dimension of work relationships, particularly when the task interdependence between project managers and developers is high. It is not surprising how closely project managers and developers work in a coordinated way as both parties desire to obtain project goals. Across the six cases in sample, the types of task and people interdependence between project managers and developers are similar to Napier et al.'s (2009) research, include planning (e.g., create schedule and estimates, task breakdown, task prioritization, scope), control (e.g., project status and issue tracking), team composition, system development (e.g., defining features), client management (e.g., user requirement, interaction with customers), problem solving (e.g., technical or interpersonal issues), and deliverables (e.g., documentation for customers). For instance, P10-PM and P10-DEV worked together on project planning through regular meetings.

> *"We did a lot of joint planning between us and Vendor at the beginning. And P10-PM was involved in that quite a bit. So it was him and I and the other PM, mostly, in meetings for that. So we worked on that quite closely. He certainly had a very good grasp of all the different aspects needed to go into the plan and sort of push to make sure that those interests were accepted by Vendor and put into the plan [including work breakdown and schedule, and the nitty-gritty details of this project."* (P10-DEV)

P3-PM and P3-DEV dealt with potential customer issues via discussion.

> *"On [the mobile application] project we felt that the timeline was very tight for one particular deliverable, and what worked really well was sitting down with P3-DEV, coming up with options to present to the client, and agreeing on the risks of each and how we would manage if the client picked the first option versus the other, and then presented it together to the client." (*P3-PM)

P7-PM commented on his working experience with project members who have different mental models.

"…everyone has their own styles, right, so it kind of comes with the territory, I guess. You have to adapt to how other people manage their own work. I mean, you get into certain situations where you disagree with someone, but you have to kind of figure out how they like to do things and accommodate that…maybe someone needs a few more reminders to get something done, or maybe you need to set up an actual meeting with them to go over something and get it finished on time, that kind of stuff…[you need to] adapting to the best way to do things." (P7-PM)

The above coordination illustrates two prominent coordination processes: explicit coordination (P3 and P10) and implicit coordination (P7) (Espinosa, Lerch, & Kraut, 2004; Stout, Cannon-Bowers, Salas, & Milanovich, 1999). Explicit coordination relies on plans, rules, standards, procedures, communication, and basically any coordination where coordination parties' intention and behaviors are articulated. Conversely, implicit coordination transpires when coordination parties carry out coordination action without explicit mechanisms or overt communication. Coordination processes do not necessarily lead to coordination success. However, in this study, participants mostly discussed the effective coordination experience. Because I am not able to assess deliverables or outcomes of each coordination incidents, I call the positive outcome of coordination as perceived effective coordination.

In the following sections, I will report implicit and explicit coordination processes and relate them to my first and second research questions: how do similarity and understanding of ISDP mental models influence work relationships, particularly effective coordination?

4.5.1.1 Understanding, Similarity, and Implicit Coordination

Implicit coordination takes place when one can "anticipate the actions and needs of their colleagues and task demands and dynamically adjust their own behavior accordingly, without having to communicate directly with each other or plan the activity" (Rico et al., 2008). It reduces communication overhead, conserve cognition resources, and expedite coordination efficiency (Entin & Serfaty, 1999; Mathieu et al., 2000; Serfaty, Entin, & Johnston, 1998). For instance, P1-PM invited P1-DEV to work in a new internal project because of the benefits of potential implicit coordination.

"So it was a lot easier for P1-PM to just get to the point and be, like, "Hey, this is the focus. This is the business value." Because I understood him, I knew what questions to ask and basically what we needed to do to get this off the ground." (P1-DEV, the high understanding and high similarity condition)

As indicated in the definition, adaptation lies at the heart of implicit coordination. Given differences in knowledge, beliefs, and preferences, project managers and developers enact different strategies and behave differently in coordination. To ensure effective management of interdependencies, either party (or both parties) needs to adapt and work out agreed-upon ways. Adaptation rests upon understanding so as to predict other's intentions and actions (Burke, Stagl, Salas, Pierce, & Kendall, 2006). For instance, P3-DEV demonstrated a high level of understanding (as did other participants in the high cross-understanding condition).

"I feel like I have a pretty good understanding of why she makes certain decisions without having to ask her to explain herself." (P3-DEV, the high understanding and low similarity condition)

In analyzing data, different behaviors charactering adaptation emerged. One frequently mentioned behavior is tailoring interaction and providing unsolicited project-relevant information based on the understanding of mental models (P1-DEV, P3-PM, P6-PM, P6-DEV, P7-DEV, all in the high understanding condition). P6-DEV indicated his understanding of P6-PM's ISDP knowledge.

"I understand his management style [status update with precise information]. I know how to get it done from him so if I need something...once you know his management style then you will modify your questions accordingly and you will do your preparations before going to him. Then it will be quick and effective." (P6-DEV, the high understanding and low similarity condition)

Adaptation affects coordination effectiveness and perceived work relationships. P6-PM made the following comments on how helpful P6-DEV is in comparison with other project members.

"P6-DEV knows the information that I'm looking for even without me asking. So I think he really does a great job there giving me all the information I need, presenting it in a way that it's clear and right there, that I don't need to go through an email of 50 pages before I find one small question." (P6-PM, the high understanding and low similarity condition)

Another salient behavior is the adaptation of one's way to manage project to meet his/her co-worker's knowledge and preferences (P1-PM, P3-PM, P3-DEV, P6-DEV, P7-PM, all in the high understanding condition). P3-PM summarized the importance of an understanding of mental models on the adaptation.

> *"I think it [understanding] helps a lot. I believe if I had no knowledge of P3-DEV whatsoever, I would go blindly in my project management methodology and may find either resistance to certain things or a lack of collaboration due to misunderstanding or just because of dislike. So I think it has a very large impact."* (P3-PM, the high understanding and low similarity condition)

For instance, P3-PM knows that P3-DEV *"prefers planning while doing versus then just doing a whole lot of planning in advance"*. P3-DEV later confirmed accuracy of P3-PM's understanding and appreciated the way they worked together.

> *"So I tend to be a bit more reactionary in my methodology. And in working with P3-PM, she seems to be happy enough working in that way. So when I say we don't spend a lot of time in the issue tracking tool, just enough time to have a framework to kind of get started… I think her as well as I prefer to have the developers sort of… here's your feature in the issue tracking, and I'm not going to break it up for you. You break it up yourself, and you create the subtasks however you feel fit to do, right? And P3-PM is very happy to let developers do that."* (P3-DEV, the high understanding and low similarity condition)

An understanding of knowledge and preferences also helps adaptation behaviors on task assignments. Some project managers adjusted task assignments to align developers' preferences (P1-PM, P3-PM, P7-PM, both in the high understanding condition).

> *"I know more what stresses P1-DEV [customer involvement and interaction], what he likes or doesn't like, the type of communication, yeah, I mean what he's comfortable with, and I can adapt the tasks or the assignments that I give him."* (P1-PM, the high understanding and high similarity condition)

Interestingly, adaptation behaviors are demonstrated in the high cross-understanding condition irrespective of similarity. Theoretically speaking, if project managers and developers have similar ISDP mental models, adaptation is not necessary because they may work in a coordinated way naturally. However, the results suggest that even when people have similar knowledge, beliefs, and preference, adaptation happens. This can be due to the fact that perfect similarity is unlikely, if not impossible. It is also conceivable that the interaction dynamics creates unexpected events for people who have similar mental models to respond differently. The data from the high understanding cases suggests that adaptation occurs constantly when an accurate understanding exists.

On the contrary, I found that a lack of understanding inhibits implicit coordination as I compared high understanding cases with low understanding cases (P2 and P10). For instance, P2-PM did not possess a good understanding of P2-DEV's mental models.

> *"I've worked with P2-DEV for probably six months but it's not on a day-to-day basis. So I'm still gauging, kind of. I know him better than a lot of people here but it's not like I'd be able to tell you exactly how he'd react in a new situation."* (P2-PM, the low understanding and high similarity condition)

A lack of understanding causes some issues. One of incidents is P2-DEV's interaction with an important sponsor of the projects in a social setting. He talked about project issues still under discussion. It appears that P2-DEV did not hold the concept "management of sponsors and champions" and confused the concept with "project transparency". Due to a lack of understanding, P2-PM devoted more efforts to explicit coordination.

> *"I'm careful on how I communicate and I try and be clear with P2-DEV. So sometimes I do need to spend more time working through things with him."* (P2-PM, the low understanding and high similarity condition*)*

Another pair (P10) in the lower cross-understanding condition also did not demonstrate any implicit coordination. It is worth noting that a lack of implicit coordination does not suggest ineffective coordination and work relationships. Different modes of coordination, explicit or not, can still contribute to perceived coordination effectiveness and work relationships. For instance, P10-DEV showed her knowledge and beliefs in explicit coordination mechanisms implemented in her organization (FinanceCo), that is, a mature ISD project management and governance structure, including a set of documentation, change management processes, division of labor, and so on.

> *"In our conversion project, it might've been something… once we got into more detail on a particular piece of development, everybody understood more and they say, "Oh, okay, if we have to satisfy this requirement, it's going to be six months of development. Do we really need to satisfy this requirement?" So we had a process for that decision-making, which I was responsible for coordinating, where it came back to our steering committee – the appropriate people at that meeting – to make a decision. "Do we do the development? Do we change our product and process?" How many did we have? I think we ended up with 42 or 43 changes to our product as a result of this. We have them all nicely documented; everything is agreed to and communicated. So that's quite a bit of churn to happen throughout a project but we managed it very, very well because we had a specific process to deal with it and it was a great process."* (P10-DEV, low understanding and high similarity condition)

She and her project manager work together well by meetings, documentations, and evaluation procedures. This suggests that different modes of coordination, both explicit and implicit ones, all can be of value. Nonetheless, their contribution to coordination effectiveness varies. Prior literature has showed that the coordination mode exerts its influence when it fits the nature of tasks and task environment (Espinosa et al., 2004; Fiore, Salas, & Cannon-Bowers, 2001).

With regard to implicit coordination, people can benefit from implicit coordination when they cope with complicated tasks which require understanding the intricacies of projects (Espinosa et al., 2004). Under such circumstances, explicit coordination would be costly (e.g., documentation of large and complex software project) and would not allow for transfer of tacit messages (e.g., politics issue in client organizations). Furthermore, implicit coordination is beneficial when coordination needs to be done in the short span of time (Entin & Serfaty, 1999). Because people are aware of others' knowledge, beliefs, and preferences, they can frame questions, propose solutions, or react more quickly. Dissimilar to research that has been done in other contexts, such as cockpit coordination (Orasanu, 1994) and military combat coordination (Blickensderfer, Cannon-Bowers, & Salas, 1998), software development may not need coordination in such a timely manner. Nonetheless, in some contexts, especially when people are involved in coordination, coordination partners need to respond promptly. For instance, if implicit coordination existed, a project manager and a developer in aforementioned P2 case would have communicated in a coordinated way (i.e., framing messages in a candid but appropriate manner) when the sponsor asked "how are you finding things?"

I will leave the discussion of explicit coordination and its boundary conditions to the next section. Table 10 summarizes the results discussed in this section. It provides evidence that the level of accurate understanding is related to implicit coordination.

Table 10 Result Summary - Implicit Coordination

		High understanding and High similarity		High understanding and Low similarity		Low understanding and High similarity	
		P1	P7	P3	P6	P2	P10
Implicit coordination	Unsolicited project-relevant information	Yes	Yes	Yes	Yes	No	No
	Adaptation to others' ISDP knowledge and beliefs	Yes	Yes	Yes	Yes	No	No
	Task assignments	Yes	Yes	Yes	No	No	No

4.5.1.2 Explicit Coordination

Different from implicit coordination with tacit anticipation and adaptation, explicit coordination is used purposefully. Prior research has paid special attention to two modes of explicit coordination: impersonal and interpersonal (Ellwart, 2011). As the analysis went along, the two modes of coordination came out frequently. Although I did not spot the systematic pattern among understanding, similarity, and the impersonal mode of coordination (this is likely due to the focus of the interview protocol), there is a potential linkage between understanding and the impersonal mode of coordination. I also identified a new theme, the compatibility of ISDP mental models, which is helpful for perceived coordination effectiveness. Emergent findings warrant attention to the impersonal mode of explicit coordination.

Impersonal mode of coordination. The impersonal mode of explicit coordination has been at the center of discussion from the traditional organization perspective because it provides control and reduces uncertainty for coordination. This type of coordination fits well for routine tasks and large teams (Espinosa et al., 2004). Different terms, such as administrative mechanism (Faraj & Sproull, 2000) and programming (March & Simon, 1958), characterize the impersonal mode as a structured mechanism governing coordination behaviors and outcomes. The common practices include defined procedures, standards, planning, project documents, design inspections, review meetings, and retrospective meetings. For instance, project managers and developers in SWCo (P1, P2, and P3) adopted Scrum practices that are effectively used to prioritize important tasks, track project status, break down tasks, and ensure transparency.

> *"We followed Agile on that project. We had a daily scrum. We had sprint planning every two weeks. We broke up consumable-sized features every two weeks in our sprint planning and assigned them to developers. We were held accountable to patch a day."* (P3-DEV, the high understanding and low similarity condition)

Standards and templates are another prevalent coordination mechanism as both the project manager and the developer of P7 indicated below.

> *"I find if you have a well-developed [template] program and you adhere to certain quality standards, it helps in a lot of ways. It helps people understand your program if you for whatever reason leave the project. It helps with maintenance down the road. It's got tons of advantages."* (P7-PM, the high understanding and high similarity condition)

A variant of the impersonal mode of coordination is coordination technologies, such as Computer Aided Software Engineering (CASE) tools and project management tools. These technologies support the aforementioned explicit coordination mechanisms for different objectives. On the one hand, the tools help control and enforce the predefined coordination mechanisms (Henderson & Cooprider, 1990; Whitehead, 2007), such as configuration management, issue tracking, and version control. Facing challenges in the management of issues, P7-PM expressed his preferences to manage them using an online tracking system.

> *"the centralized issue and risk-tracking tool is really essential…you can track issues that aren't in someone's email inbox because that can get super risky, right, because people forget emails, and you don't have visibility on someone's email inbox whereas a centralized tool like this you kind of know where everything stands."* (P7-PM, in the high understanding and high similarity condition)

On the other hand, the tools facilitate project members to effectively communicate and coordinate different planning and design tasks (Favela & Peña-Mora, 2001). For example, P6-PM mentioned a forecast tool used to improve planning activities in which he and P6-DEV worked together.

> *"We're just getting the tools to get better at it, but I wish I had them [forecasting and planning tool] like four months ago, and I still think they could be better, more real time, or the ability to create quicker, like, what-if scenarios. Like, what if we add another resource here? Will that help us in three months from now? Stuff like that."* (P6-PM, the high understanding and low similarity condition)

When I compared cases, the first thing that came to attention was the different impersonal mode of coordination adopted by organizations. Although many coordination mechanisms are universal, such as scheduling of tasks and releases, standardization of templates, task decomposition, and issue tracking, these mechanisms can be implemented in different ways. Take coordination mechanisms during planning for example. SWCo (P1, P2, and P3) embraces Agile methodologies and applies short iteration planning, which requires more interaction between project managers and developers over the course of projects as compared to the plan-driven approach (or so-called Waterfall software development life cycle), which is still the main development approach in ConsultCo (P6 and P7) and FinanceCo (P10), with an assumption that complete requirements can be obtained at the beginning and are required to pay much attention to a plan and design. Templates have been used in ConsultCo and FinanceCo to facilitate heavy upfront planning and design (e.g., project notebook outlines, meeting agendas, Request for Proposals (RFP)) and to mitigate risks of possible changes after planning (e.g., candidate risk events lists and detailed risk mitigation plans for a specific risk event).

Then, it became clear to me that perceived coordination effectiveness is contingent upon whether project managers and developers agree on coordination mechanisms. This state can be called the compatibility of ISDP mental models with the impersonal mode of coordination. Compatibility is different from similarity. Project managers and developers do not need to possess the exactly same concepts suggested by the coordination mechanisms. More importantly, they need to understand the mechanisms and appreciate their value. In my samples, all participants show their understanding of the value of existing coordination mechanisms and believe that they help with coordination effectiveness. Despite the current compatibility among the two actors and coordination mechanisms in each case, case P6 (the low similarity and high understanding condition) had incompatibility in the early days of coordination. P6-DEV mentioned that P6-PM comes from a different consulting company and believes in different processes and methods. P6-PM did not like some templates and processes and consequently made some changes, which caused some troubles for P6-DEV because they were different from the mechanisms that he had been using in other projects. Ultimately, P6-DEV adapted to the revised coordination mechanism, resulting in the compatibility state.

Similar to the findings of this study on the compatibility of ISDP mental models, prior literature also supported the notion of compatibility and suggested that organizations should adapt to different coordination mechanisms that align the project context with, not only the characteristics of projects but also the knowledge and beliefs of project members (Karlsson & Agerfalk, 2009; Pikkarainen & Passoja, 2005).

The second emergent theme for the impersonal model of coordination is coordination mechanism tailoring. Some project managers saw the deficiencies of the existing mechanisms, so they included new coordination mechanisms or tailored the existing mechanisms to ensure coordination effectiveness in their projects. For instance, P7-PM introduced daily scrums in a project to resolve project transparency issues. P10-PM adopted Kanban board, which is complementary to traditional project management tools, to present and manage work items more dynamically. P3-PM tailored the existing Scrum practices and posited the positive impacts of the improvement.

> *"…to be more efficient as a team and to work better and to always look at being as lean in process as needed and to adapt… we almost changed from an agile to more of Kanban approach to cater to the client's changing requirements and priorities and such. All of these came through continuously being together [retrospective meetings] on the project and talking out how to make things better."*(P3-PM, the high understanding and low similarity condition)

This is different from tweaks of the way of managing projects to a co-worker that I illustrated in the implicit coordination. In this situation, the coordination mechanisms are revised for the whole project team rather than one individual co-worker. In doing so, project managers should take project members' knowledge, beliefs, and preferences into consideration. Otherwise, new coordination mechanisms may cause tension. Meanwhile, the success of changes requires most of developers to understand the intention of changes. Therefore, cross-understanding should still contribute to the process of restructuring coordination mechanisms. The success of changes may have impacts on work relationships. On the one hand, a project manager builds a better work environment where individual developers can work effectively. On the other hand, even though the developer may not have specific preference for new coordination mechanisms, their perceptions of the project manager would be influenced by other developers' judgment (Klimoski & Donahue, 2001). That is, the developer may form positive perception of his/her project manager if other developers accept coordination mechanisms and vice versa.

I do not have accounts from project managers about the process of changes and observations from developers regarding how changes in coordination mechanisms directly affect coordination effectiveness. Given resistance to different mechanisms often reported in popular press, this line of argument deserves further investigation.

Table 11 shows that all pairs have compatible mental models with the existing explicit coordination mechanisms and have good perceived coordination effectiveness.

Table 11 Result Summary - the Impersonal Mode of Explicit Coordination

	High understanding and High similarity		High understanding and Low similarity		Low understanding and High similarity	
	P1	P7	P3	P6	P2	P10
Compatibility of ISDP mental models	Yes	Yes	Yes	Yes	Yes	Yes

Interpersonal mode of coordination. The interpersonal mode of coordination is all about communication. It is useful when administrative coordination is lacking and the complexity of coordination items is high (Espinosa et al., 2004; Van de Ven, Delbecq, & Koenig Jr, 1976). Through communication, coordination parties engage in more personal and intensive exchange of information. Communication serves for different coordination situations between project managers and developers. The data suggests three types of communication: information exchange, feedback seeking, and knowledge integration.

(1) Information Exchange. At the foundation level, the interpersonal mode of coordination is convened with the exchange of information. Such exchange ensures a shared understanding of projects and maintains bonds between project managers and developers.

"Every week we have a tech status meeting. So P7-PM tries to focus on each item and discuss with each individual team member: what's the status on this item? This is the due date, right? This is the deadline. How are we handling this? If we are waiting on something, we need to follow up. ...at the end of this [tech status meeting] sometimes he invites me in his office and talks about… for example, if we are talking about one item, development and what's going on with this, so how we are going to handle it within the next few days, what approach we should take, if we need to talk to some other team member, if we are waiting on something. So that's all we discuss, like, in a friendly manner, or informally … whenever I leave for home every day, I go to his office and talk to him. So he says, "How it goes today?" I say, "Oh, this is how it's going. This is what we did today, and this is what we are planning to do within the next couple of days." So it's informal, but he gets to know what's going on." (P7-DEV, the high understanding and high similarity condition)

In the sample (all of the six cases), communication between a project manager and a developer is characterized by openness, honesty, and transparency. I found that trustworthiness[25], positive expectation of trustees (Mayer et al., 1995), is the major drive that helps information exchange. Particularly, benevolence, one of the three characteristics of trustworthiness and a belief that "a trustee wants to do good to the trustor" (Mayer et al., 1995), is concomitant with effective information exchange. Both project managers and developers show caring and supportive motives.

"...I cared [P1-DEV], he saw that I care, so he opened and then that created a trustworthiness that enabled much more frank discussions and more result-oriented discussions." (P1-PM, the high understanding and high similarity condition)

Furthermore, benevolence trust creates emotional bonds that bind trustors and trustees (McAllister, 1995), which in turn allows them to effectively and easily communicate difficult topics, such as personnel issues (P1, P3), low quality of deliverables (P1, P6), and major project challenges (P3). For instance, P6-PM indicated his knowledge of P6-DEV at the personal level helps provide feedback on deliverables.

[25] Researchers have distinguished trust into two components: the intention of accepting vulnerability and positive expectation (Colquitt, Scott, & LePine, 2007). Trustworthiness is about positive expectation and usually considered an antecedent of trust.

"it's much easier to communicate. You can be more honest with people… If I don't know anybody at all – I only work with them – and then all of a sudden I have to tell them, "Your spreadsheet looked like crap," that person might get really offended. But if I can kind of butter it up and say, "You know what? I know your work otherwise, and I know you're a good guy, whatever, but this spreadsheet just needs work." Then it becomes a more balanced message back to P6-DEV." (P6-PM, the high understanding and low similarity condition)

P6-DEV added to reciprocal benevolence between him and P6-PM.

"if you compare me with the other cohorts who are directly reporting to P6-PM…I don't have that fear or that wall in front of me. If you look at other coworkers who are reporting to him they don't have that much freedom. They don't feel that comfortable in going and talking to him very easily. I would say that helped me a lot." (P6-DEV, the high understanding and low similarity condition)

(2) Feedback seeking and giving. Feedback seeking and giving is another interpersonal coordination behavior that requires benevolence trust (P1-DEV, P2-DEV, P3-DEV, P6-DEV, P7-DEV). Prior research found that project members who are free to express their doubts and ideas contribute to better coordination outcomes, leading to better products (Lovelace, Shapiro, & Weingart, 2001). A lack of feedback seeking behaviors prevents learning opportunities and project success. Despite the benefits of feedback, people are reluctant to ask for help because they feel vulnerable if they admit their lack of knowledge of complete tasks (Argyris, 1982). P3-DEV pointed out the importance of feedback seeking and benevolence trust.

"I've seen people sit on a problem for a week because they don't want to admit they don't know how to fix it… if somebody can make that adjustment here [in SWCo, it is believed that people are caring one another], then it works well because what you end up seeing is somebody sitting on a problem for four hours and then raising the flag and saying, "Look, guys. I can't figure out how to do this." And then you've got a team of five or a team of 10 all looking at the same problem, and then usually within a couple hours you're unstuck and you're moving forward. Everybody's happier, and the product moves forward faster. And the trust comes in in that your project manager, for example, is not going to look at you and go, "Oh, well, P3-DEV couldn't figure out

how to make that list look the right way last week, so there's a mark against him." You know what I mean? Like, if you have the trust that admitting when your skills are failing you, if you have the trust in your coworkers and your project manager to raise the flag, then everyone can go home earlier and fix the problem." (P3-DEV, the high understanding and low similarity condition)

(3) Knowledge integration. Communication can escalate from information exchange into more intensive knowledge exchange and knowledge integration which are beneficial to both business and technical problems in ISD projects (Tiwana, 2004). Diverse skillsets and perspectives are needed to address such complex issues (van Knippenberg, De Dreu, & Homan, 2004). Integration of diverse knowledge is difficult because experts who have different beliefs and preferences frame problems differently (Cronin & Weingart, 2007). The understanding of ISDP mental models helps remove the barriers. It builds common understanding on the relative importance of others' goals. For instance, P1-DEV was aware of P1-PM's dominant ISDP belief - customer.

> *"…certain things value to him more than other things, especially when you're working on a project with him… customer satisfaction is one thing, right. So you speak to him regarding how we can work towards that. He's going to value a lot… value that kind of discussion a lot more. He's going to provide you more feedback. He's going to provide you with support… any feedback you would need to help him accomplish that goal."* (P1-DEV, the high understanding and high similarity condition)

On the other hand, P1-PM showed appreciation for the belief of product (though it is not his dominant belief. Similarity index: 0.06) when P1-DEV indicated its importance. Cross-understanding helps the integration of expertise of P1-PM (business strategy and planning) and P1-DEV (mobile development).

> *"P1-PM values that kind of input [a very specific technical reason why we experienced this issue]. So you give him that and he builds on top of that. And he's, like, "Okay, that's good. What if we drop this feature altogether and try to build a different feature because in the long run, this is what the customer would value even more?" … I've had consultation like this with him. We sit there and we brainstorm. We're, like, okay, that's cool. And then what about this? And then this is*

how it might be… these are the technical obstacles in the way to accomplish this. And then we'd go back and forth basically, building on top of each other's ideas and feedback." (P1-DEV, the high understanding and high similarity condition)

The project manager and developer of P7 were well aware of one another's ISDP mental models regarding knowledge in collaboration, control, documentation, and technical excellence along with an accurate understanding of one another's beliefs in team and product. Given a good understanding of each other, they were able to tap into one another's expertise (P7-PM's expertise on ERP and P7-DEV's expertise on mobile development) for a presentation of new technology. They exchanged knowledge frequently, integrated each other's knowledge, and made a well-received presentation.

When I compared the cases of P7 and P10 (the high similarity and high understanding condition) with the cases of P2 and P10 (the high similarity and low understanding condition), the similarity of mental models did not contribute to integration of diverse expertise. That is to say, the mere presence of similarity is not sufficient. Project managers and developers need to be mindful of what their co-workers know and believe. Accordingly, their conversation and discussion would be more productive.

Table 12 summarizes the interpersonal mode of explicit coordination. There is no difference among the three conditions for information exchange and feedback seeking and giving. However, benevolence trust is the necessary condition for these two behaviors. For P3 and P6, I did not identify any feedback seeking behaviors. This can be due to the seniority of the developers (P3-DEV has 15 years of ISD experience and P10-DEV has 14 years of ISD experience). With regard to knowledge integration, it shows that the higher understanding condition generally facilitates the process of knowledge integration. For P6 who did not demonstrate knowledge integration behaviors, the potential explanation is that P6-PM and P6-DEV share similar expertise (i.e., technical aspects of ERP) and may not have sufficient diversity to stimulate knowledge integration.

Table 12 Result Summary - the Interpersonal Mode of Explicit Coordination

		High understanding and High similarity		High understanding and Low similarity		Low understanding and High similarity	
		P1	P7	P3	P6	P2	P10
Explicit coordination (interpersonal mode)	Information exchange	Yes	Yes	Yes	Yes	Yes	Yes
	Feedback seeking and giving	Yes	Yes	No	Yes	Yes	No
	Knowledge integration	Yes	Yes	Yes	No	No	No

4.5.2 Trustworthiness

Coordination is a characterization of behaviors that transpires in effective work relationships between project managers and developers in the cases of this study. It gradually became clear that trustworthiness, which is the attitudinal aspect, also characterizes effective work relationship. When people hold positive attitudes toward their co-workers, they have positive expectations that they can count on their co-workers for fulfilling their needs in the work context.

Benevolence Trust. In the previous section, I have shown the relationship between benevolence trust and interpersonal mode of coordination. Benevolence can be considered as affect-based trust (Colquitt, LePine, Piccolo, Zapata, & Rich, 2012; McAllister, 1995). It is important in the sense that both parties during the exchange would take each other's needs and desires into account. The expectations of positive reciprocal consequences increase a sense of attachment and belonging at work.

Competence Trust. Competence as another aspect of trustworthiness deserves attention. In the highly task-interdependence context like ISD projects, whether one holds positive expectations about his/her co-worker's competence and skills plays a significant role in the perception of work relationships. P2-PM pointed out the importance of competence trust in work relationships.

> *"I think that there needs to be trust. I need to have respect for the person. And to me, they need to show that they know what they're doing."* (P2-PM, the low understanding and high similarity condition)

While competence trust permeates across all six cases, its role in work relationships is less dominant when cross-understanding increases. The relationship between trust and understanding is delicate. Trust is "somewhere between total knowledge and total ignorance" (Simmel, 1964). If one has a full understanding of another, trust does not need to exist. Conversely, if one does not have any understanding of another at all, expectations are less likely to be formed. Competence trust is initially developed from an understanding of expertise, that is, a general understanding of co-workers' competence and capabilities. For instance, P10-PM has a general understanding of P10-DEV's expertise. However, the quote below does not show that he knows much about the way P10-DEV deals with requirements and users.

> *"I respect P10-DEV professionalism and she's got the knowledge and I rely on her…how she addresses it is up to her. Like do you want to interviews, what facilitation session, who to bring in, etc., etc. Up to her to work through that."* (P10-PM, the low understanding and high similarity condition)

Similarly, P10-DEV was not aware of her project manager's mental models but she trusted her project manager's ability to manage complex projects.

> *"P10-PM is obviously very experienced in that complexity of the project because for him, it seemed quite easy."* (P10-DEV, the low understanding and high similarity condition)

It is conceivable that if the degree of understanding of one another does not increase, competence trust would be the major factor to form perceptions of other people and affect the work relationships. Even when individuals need to understand another in order to make a good judgment on potential consequences (e.g., in coordination tasks involving high risk where deliberative thinking takes precedence over heuristic thinking) (Fiske, 1993), they do not have much information to process and often succumb to their expectations of other people first, which could be trust or distrust.

For the pairs in the low understanding condition (P2 and P10), they mentioned competence trust more frequently than those in the high understanding condition (P1, P3, P6, and P7) (8.5 and 2.5 code segments per pair respectively). The difference does not mean that competence trust is higher in the low cross-understanding condition. Instead, it implies that the perception of effective work relationship constitutes competence trust primarily. They are uncertain about prospective action of others. Therefore, less implicit coordination would occur (P2 and P10 did not mention any implicit coordination behaviors).

The Relationship between Competence Trust and Benevolence Trust. In the work setting, competence trust also affects benevolence trust because the basis of interaction is predominantly the task (McAllister, 1995). Professional respect engenders positive feelings and therefore benevolence trust increases (P1-DEV, P3-DEV, P6-PM, P6-DEV, P7-PM, P7-DEV).

> *"P7-DEV is very trustworthy. They [Customers] know he's a hard worker, and they know he does good work, so it's good for him, and it's good, obviously, for ConsultCo because we want our clients to have trust in us kind of thing, right? I mean, the guy is up for any challenge, really. I haven't asked him to do anything that he hasn't excelled at, you know what I mean? He's really, really good."* (P7-PM, in the high understanding and high similarity condition)

Consistent with social exchange theory, their work relationships initially contained task-related goods, such as information and advice but gradually started to include affect-related goods, such as caring, support, loyalty, and commitment. With trust from his project managers, P7-DEV received substantial support and help and described his commitment to the relationships

> *"If P7-PM helps me out, right, in difficult times [as P7-DEV mentioned earlier "P7-PM helped me a lot in understanding the process, in understanding the technology"], I will do everything for him, you can say."* (P7-DEV, the high understanding and high similarity condition)

Such high benevolence trust creates a virtuous cycle of positive work relationships.

Sources of Benevolence Trust. The data suggests that in addition to competence trust, there are two more sources for benevolence trust. One is interpersonal citizenship behaviors (ICB) (see next section for discussion) while the other one is an understanding of social aspect of the co-worker. The personalized knowledge about co-workers provides more social data for one to make an attribution and build benevolence trust (P1, P3, P6, P7).

> *"Basically, personally I want for work with someone who cares. I wouldn't work for someone that… It's nice to say, "How is your weekend?" but I feel that they genuinely care about what I'm going to answer back. And to the point where instead of just "How's your weekend?" it's maybe even "How's Valerie?" or I'll go, "How's Phoenix, your newborn? Is it six weeks?" or… You know, there's that level of caring where if I genuinely care for that individual, and first off there's their trustworthiness is a lot higher, the ability to talk straight is a lot higher, the ability to be understanding is a lot higher."* (P1-PM, the high understanding and high similarity condition)

> *"Well, I find for me I think the key element is I can relate to the person on a human level, right? So projects are going to come and go, and they're always going to be different. You're always going to have a different team. But, to me, to be able to work well with somebody, I need to be able to know… I need to be comfortable just sitting in a room chatting with them over coffee about anything, whether it's the project or whatever the episode of the show you watched last night or anything, right? And it's also very important that I need to get a genuine feel for the person, feel that they're genuine when they're speaking. I don't want to feel… nobody wants to talk to somebody where they feel like the person is just telling you what you want to hear or being condescending when they're talking to you or stuff like that."* (P3-DEV, the high understanding and low similarity condition)

Table 13 shows that all pairs have a high to very high level of benevolence and competence trust except P2. P2-PM showed high competence trust in P2-DEV's expertise in architectural design and programming but expressed concerns about his expertise in other areas, such as client management. The understanding of ISDP mental models and similar mental models do not indicate differences in both benevolence trust and competence trust. Table 14 further presents the influence of different sources of understanding on benevolence and competence trust.

Table 13 Result Summary - Trustworthiness

	High understanding	High understanding	Low understanding

	and High similarity		and Low similarity		and High similarity	
	P1	P7	P3	P6	P2	P10
Benevolence trust	Very high	Very high	Very high	High	High	High
Competence trust	High	Very high	High	High	Moderate to high	Very high

Table 14 Relationships between Understanding and Trustworthiness

	Understanding of Expertise	Understanding of Social Aspect	Understanding of Individual Needs and Constraints
Benevolence trust	N/A	P1, P3, P6, P7	N/A
Competence trust	All cases	N/A	N/A

4.5.3 Beyond Job Roles

For highly effective work relationships, I observed helping behaviors beyond their job requirements, which is referred to as interpersonal citizenship behaviors (ICB) (Settoon & Mossholder, 2002). ICB is highly correlated to effective work relationships and also beneficial to individual and project productivity by enhancing job satisfaction, commitment, and positive work environment (Podsakoff, MacKenzie, Paine, & Bachrach, 2000). ICB can be either person-focused by providing "self-esteem maintenance and deals with problems of a more personal nature" (P1-PM, P3-PM, P3-DEV, P7-PM) or task-focused by offering help to complete tasks that are not one's responsibility (P1-PM, P2-PM, P3-PM, P1-DEV) (Settoon & Mossholder, 2002). For example, P1-PM expressed his concern about P1-DEV's well-being, offered counseling, and adjusted tasks.

"I think P1-DEV was very, very hard on himself. Like he was starting his career and he wanted to pretty much be perfect…like he was too stressed. So we had a lot of backs and forth on that. So for instance if a customer asked for this thing, then he would kind of panic and would kind of be frozen by the potential consequences instead of kind of taking it step by step… [I] provide some coaching and more of almost a personal level to say, "Don't worry," basically… Like P1-DEV had difficulties sleeping at one point and it clearly impacted his performance, but there were

ways that we could play around that and move along around that." (P1-PM, the high understanding and high similarity condition)

Additionally, even though P1-PM's KPI has nothing to do with P1-DEV's career progressing, he talked to P1-DEV frequently. He worked closely with the development managers to achieve P1-DEV's career goal.

"I did ask P1-DEV, "Where do you want to go? What do you want to achieve?" and based on that, then I adapted the roles and the tasks and the assignments to try to meet… like a win-win, right? So I mean the project wins and the individual wins, because he can achieve through the project his personal aspirations." (P1-PM, the high understanding and high similarity condition)

P1-DEV appreciated all his support and saw him as a role model.

"P1-PM would identify basically who needed more guidance, who needed more support, who did he need to basically keep an eye on. I don't want to say it… like, make it sound like that a person needed more help or needed more support. It's just he felt that perhaps it would be beneficial in the long-term if I did give this guy more support. Maybe he wants to advance and build his skill sets and get more exposure or needs that communication bond. So he would check in, because I was inclined to… like, learning more project management from day one. So I really… I found that very beneficial. I expressed that to him. I'm, like, "I really enjoyed that you check up on me, because I want to build my skill sets." So I would learn a lot from him." (P1-DEV, the high understanding and high similarity condition)

P1-DEV has been promoted to a project manager and he attributes his skills in project management, including communication, team development, client management, and mentoring, to P1-PM's coaching and helping.

ICB is often reciprocal. P2-DEV received help from P2-PM when dealing with customers who have strong personalities and have strong views on religion. Meanwhile, when P2-PM worked in other projects and required some advice, P2-DEV offered assistance.

"P2-DEV doesn't have BB10 [BlackBerry OS 10] experience either." But he came in and with his architecture background, a diagnostic of the technology, he was able to say, "This is how it needs to be put together," and helped coach the team to what they needed to build – not how they were

going to build it – but what they needed to build from a technical perspective to make sure that things flowed from end to end." (P2-PM, in the low understanding and high similarity condition)

The Relationship between Benevolence Trust and ICB. Consistent with prior literature (McAllister, 1995; Settoon & Mossholder, 2002), the results show that ICB is closely related to benevolence trust (P1, P2, P3, P6, P7) and often comes from people who hold more resources (P1-PM, P3-PM, P6-PM, and P7-PM). The data also suggests an accurate understanding of needs and constraints would help with ICB (P1-PM, P3-PM, P6-PM, P7-PM, P7-DEV). The understanding of needs and constraints is knowledge about the co-worker's personal aspirations, goals, and personal issues. P7-DEV suggested the importance of an understanding of needs and constraints in addition to an understanding of expertise.

"So I guess you have to know everyone at a personal level, right? If we are working together, I should know you very well so that I could understand you. Like, what are your strengths? What are your weaknesses? Maybe I need to understand about your family, if you have any family issues going on or if you have stress, right? So that affects work." (P7-DEV, in the low understanding and high similarity condition)

For instance, P3-PM understood personal constraints of P3-DEV to discuss work matters after four o'clock and she catered to his needs. Such understanding also avoids attribution bias. She would not think P3-DEV as a "slacker" because he left early. Moreover, when P1-PM knew that P1-DEV faced development challenges due to the use of new technology, he asked senior developers to help. An understanding of needs and constraints is different from perspective taking suggested by previous literature (Settoon & Mossholder, 2002) in terms of the degree of accuracy. Perspective taking is "tendency to spontaneously adopt the psychological point of view of others" (Davis, 1983, pp. 113–114) and requires one to make inferences about others' needs. However, an understanding of needs and constraints is more tangible based on day-to-day interaction, which occurs in the highly task-interdependent project environments. In sum, the findings suggest that an understanding of needs and constraints engenders ICB, particularly when benevolence trust or previous ICB experience occurs.

There is no direct linkage between similarity/understanding of ISDP mental models and ICB (Table 15). Rather, the ICBs are concomitant with benevolence trust and understanding of individual needs and constraints (Table 16).

Table 15 Result Summary - Intepersonal Citizenship Behaviors

		High understanding and High similarity		High understanding and Low similarity		Low understanding and High similarity	
		P1	P7	P3	P6	P2	P10
Interpersonal Citizenship Behaviors	Person-focused	Yes	Yes	Yes	No	No	No
	Task-focused	Yes	Yes	Yes	Yes	Yes	No

Table 16 Relationship between Benevolence Trust, Understanding, and Interpersonal Citizenship Behaviors

	Benevolence trust	Understanding of Individual Needs and Constraints
Interpersonal Citizenship Behaviors	P1, P2, P3, P6, P7	P1, P3, P6, P7

4.5.4 Antecedents of Cross-understanding

After analyzing four cases in the high cross-understanding condition, I identified three antecedents of high cross-understanding: frequent interaction, mentorship, and anchoring events.

Frequent interaction. Frequent interaction enhances cross-understanding as it provides more evidence to form an accurate understanding. People observe and understand each other in different formal and informal incidents. For instance, P1-DEV understood P1-PM's customer-centric beliefs as well as related concepts through their interactions with customers.

"[P1-PM has] a very strategic and business mindset. So he always has an eye on the bigger picture, bigger scope of things. It's not just about this project or how I'm interacting with this customer only for the duration of this project. It's how am I interacting with this customer for… to

build this relationship with this customer, right. It's not, like, okay, from January 1st to March 31st or whatever, that's it. This is my customer, and then I'll move on to another customer. It's about building a relationship. How can I sustain this relationship, right? So he tries to enforce that as much as possible. And I saw that from, like, day one…he doesn't express that. He hasn't said that, for instance. But you can see it. He demonstrates it, right. So that's very inspiring." (P1-PM, the high understanding and high similarity condition)

Frequent interaction can also be implemented as a routine, such as daily scrums and retrospective meetings where project managers and developers interact closely (P1, P2, P3, P7). Furthermore, some project managers practiced frequent check-in and talked to developers on a daily basis (P1, P7). On the other hand, developers also took the initiative and reported what they did and were going to do (P6, P7). Informal interaction, such as team celebration (P1, P2, and P3), team get-together (P1, P2, P3, P6, P7), hang-outs after work (P7), carpools (P6), is also of considerable assistance to cross-understanding. However, whether one can actually understand other people also rests upon propensity to understand. Some interviewees expressed full trust in their co-workers' jobs and showed no interest in knowing the concepts their co-workers hold. For instance, when P2-DEV was asked about his understanding of P2-PM's way of doing planning, monitoring, and tracking, he expressed that he does not need to care about those things.

Mentorship. Mentorship creates a close bond between a mentor and mentee. Mentors contribute to mentees' career advancement, personal growth, and professional development (Ragins & Cotton, 1999). P1 and P7 have an informal mentoring relationship, which is developed spontaneously rather than assigned by organizations. Through mentoring relationships, mentees learn new skills through mentors' advice and challenging task assignments, leading to high understanding. Similar mental model is another product of mentorship as P1-DEV indicated below.

"P1-PM has influenced me quite a bit. So we do share a lot of the same principles and values… he's my first role model. He still is my role model. I can't think of anything [about dissimilar

principles between me and him].” (P1-DEV, the high understanding and high similarity condition)

Anchoring Events. Ballinger and Rockmann (2010) suggest that anchoring events, defined as memory episodes that contains extreme emotional and instrumental content, punctuate the process of work relationships and leads to change in work relationships. I believe that anchoring events are related to high understanding since the events embody rich information for one to understand another. Additionally, the extreme form stimulates people to pay attention to the events.

P1-DEV and P7-DEV both started their first job and first project at their companies. Their interaction with project managers would be anchoring events in their memory. During the interaction, they were also eager to learn from their project managers and absorb the way to implement ISD projects. The understanding of ISDP mental models is undoubtedly high. For P1-PM, although he is an experienced project manager, the project he worked on with P1-DEV is his first project in SWCo. Given his rich experience in project management, he may not change his approach to interact with developers. However, to ensure the success of his first project, it is likely that he would strive to learn more about project team members. For P3, the anchoring event is a site visit to a multinational electronic company. Additionally, this project is the first one for P3-PM. The event helps increase an understanding of each other. Since then, they have had very effective work relationships.

“…right from the start we got along great. We both had to travel down to a USA City to meet the customer and spend some time with the customer, and what I found was having that time early in the project where we were kind of away from the office and away from all of the distractions, and we could just focus on the customer and figuring out what we needed to do for this project, it went a long way in making that project very, very smooth. So things like having the opportunity just to sit and talk freely, not within a formal meeting and not talking about the project because we have to talk about it because it's 9 to 5 and we're on the clock, but just having that kind of freeform… it doesn't really happen when you're at home because when you're at home, when it's time to go home, I'm going to go home to my family, right, and she's going home to her family.

But when you have to travel for work, especially at the beginning of a project like that to meet the customer, it's very different because now you're still working your 9 to 5 or 9 to 6, 9 to 7, whatever the customer wants, but after work you go get dinner. Maybe you go for a drink. And you have a lot of opportunity just to casually talk about the project with no pressure of having to take notes or having to make decisions about anything." (P3-DEV, the high understanding and low similarity condition)

Conversely, a lack of cross-understanding can be due to a lack of anchoring events (P2 and P10). P2-PM compared her understanding of P2-DEV with her other co-workers who share anchoring events. She believed that her understanding should be enhancd significantly if she and P2-DEV shares anchoring events.

"But there's certainly a difference between working with someone who you know is good at their job but you haven't had a lot of experience with them, as opposed to someone that you've worked with day in, day out, for three years in the trenches, through difficult phone calls with the customer, through midnight testing sessions and so on. And I haven't had that with P2-DEV. I've been in scrums with him. I've been in customer meetings. I've been on airplanes with him but upon reflection, I haven't really had the really, really, bad things happen. He's not met customer expectations and I've had to work through that with him but I haven't been in a situation where the shit has hit the fan and I need to rely on him in this instance to fix something. So I think that's really where you gel as a team and sometimes you may never get to that point with someone because that opportunity doesn't arise. And that's, to me, where I would have that level of comfort that I know that they have my back and that I have theirs and we can get through anything. And I would think that one day I would get that with him but I don't think I have that right now." (P2-PM, the low understanding and high similarity condition)

Table 17 summarizes the antecedents of understanding. For the high understanding condition, frequent interaction, mentorship, anchoring events exert their influences to different degrees.

Table 17 Result Summary - Antecedents of Understanding

	High understanding and High similarity		High understanding and Low similarity		Low understanding and High similarity	
	P1	P7	P3	P6	P2	P10

Antecedents	Frequent interaction	Yes	Yes	Yes	Yes	Yes	No
	Mentorship	Yes	Yes	No	No	No	No
	Anchoring events	Yes	Yes	Yes	No	No	No

4.5.5 Summary

Figure 20 shows a model of work relationships between project managers and developers, and illustrates how understanding helps effective work relationships in terms of trust and coordination. The results indicated that the similarity of ISDP mental models does not account for effective work relationships (Research Question 1). Rather, an accurate understanding of ISDP mental models fosters implicit coordination where project managers or developers adapt their behaviors to improve coordination effectiveness (Research Question 2). Furthermore, an accurate understanding of ISDP mental models facilities knowledge integration, if diverse expertise fits the task. When lack thereof, trust would be needed to maintain positive expectations about each other. Regarding Research Question 3, in different contexts, mentoring, anchoring events, and frequent interactions, would contribute to better understanding, assuming the propensity to understand exists.

Besides the focus on an understanding of ISDP mental models, other types of understanding also help in work relationships. An understanding of individual needs and constraints makes it more likely that both parties will take on interpersonal citizenship behaviors (ICB); the relationship between these two constructs (an understanding of individual needs and constraints and ICB) is strengthened when benevolence trust is strong. Moreover, an understanding of social aspects, which is closely related to an understanding of individual needs and constraints, helps one to understand another at the personal level. Individualized knowledge usually enhances benevolence trust. Lastly, an understanding of expertise is one important source of competence trust. As one trusts another's competence and abilities, benevolence trust develops accordingly.

4.6 Discussion

ISD projects consist of project members who have different expertise and hold different ISD project knowledge. The diverse makeup provides a pool of information and perspectives for a project team to capitalize on. However, the benefits of diversity can be offset by process losses and potential relationship conflict (Webber & Donahue, 2001). This study focuses on the effective work relationships between project managers and developers who often possess diverse background and expertise, leading to different beliefs in dealing with projects. The important question to both practitioners and researchers is how project members can utilize each other's expertise and experience to achieve desirable project outcomes.

At the outset of this study, three questions were posed to guide the investigation of the above issues. First, how does the similarity of ISDP mental models affect work relationships? Second, how does an understanding of others' ISDP mental models affect work relationships? Third, what are the major drivers of an accurate understanding and misunderstanding of others' ISDP mental models? The findings in relation to these three questions are now discussed.

4.6.1 Research Question 1: The Impacts of Similarity of ISDP Mental Models

It is believed that mental models guide both one's behavior and interpretations of others' behaviors. When there are similar mental models, it is expected that the actual behaviors in various tasks between project managers and developers are likely to align with expectations. Additionally, they are more likely to expect others' intention and behavior. Conversely, differences between project managers and developers' mental models may create barriers, detachment, distance, and interpersonal conflict, thereby leading to ineffective work relationships. Nonetheless, data from the six cases suggests that the similarity of mental models does not automatically facilitate work relationships in terms of coordination. More importantly, it is an accurate understanding of others' mental models that helps interpret others' intention and behaviors. An accurate understanding of mental models is also helpful for reconciliation of interpersonal differences (this will be discussed in the next section).

Although the similarity of mental models does not directly contribute to effective work relationships, it should be noted that people who have different mental models may take longer to get an accurate understanding and may not get into the stage of high cross-understanding. In other words, in any project teams, there should be higher proportion of pairs who have similar mental models and high understanding than those who have dissimilar mental models and high understanding. P10-DEV talked about her experience in working with people who are dissimilar.

> *"If you're coming from two completely different backgrounds or beliefs, it takes a lot more work to get to the middle."* (P10-DEV, the high similarity and low understanding condition)

Data below further illustrates the benefits of similar mental models. By coincidence, one of the participants (P6-DEV) compared his work relationships with his current project manager with his previous project manager (P5-PM, the case of P5 was dropped due to a lack of task interdependence). The similarity of ISDP mental models between P6-DEV and P5-PM is very low (similarity index: 0.07 but their dominant beliefs are both enterprise-oriented). They had worked together for 8 years. P6-DEV described his difficulty in knowing P5-PM and coordination at the beginning. It took them a while to know each other after very frequent formal and informal interaction.

> *"We used to go out rarely. He would maintain his own space and I would maintain my own space. Even though I reached out, we never discussed much into details… So initially I was a bit hesitant to go and tell him my concerns [about timeframes of the project]… you don't know how your manager is and how he is going to react, whether he will support or oppose or whether he will care or will not care. It took me some time to come out of that phase and talk freely and a period of time to establish that relationship."* (P6-DEV)

However, even though the understanding increases, he perceived the work relationship with P6-PM is better than with P5-PM because of the similarity. It is unclear whether his perceived work relationship concerns benevolence trust or coordination effectiveness because I did not have data regarding trust and coordination processes between P6-DEV and P5-PM. According to the theoretical model in Figure 20, if P6-DEV holds a similarly high degree of accurate understanding towards P5-PM and P6-PM, the coordination effectiveness should be equivalent, including effective implicit coordination, advice seeking, and knowledge integration. Otherwise (the degree of understanding being low or moderate), the perceived effective work relationships could be based on benevolence and competence trust. Based on these preliminary results, I suggest that future research should investigate the transition between trust and understanding and test how the transition affects work relationships. The empirical research should strength validity and reliability of a causal relationship suggested by our theoretical model.

4.6.2 Research Question 2: The Impacts of Understanding

Our findings provide important evidence that project managers and developers who have different ISDP mental models can coordinate effectively when they have an accurate understanding of each other's mental models. Accurate understanding mitigates issues of communication challenges because by means of the process of understanding people would familiarize themselves with each other's vocabularies and terminology. This finding is consistent with Krauss and Fussell's work (1990) on mutual knowledge. However, accurate understanding has impacts beyond communication. Due to an understanding of knowledge, beliefs, and preferences, both parties are able to anticipate each other's intentions and actions and make dynamic adjustment (i.e., implicit coordination). They can effectively manage project tasks, such as project planning, monitoring, and tracking. They can further quickly adapt interaction processes so as to suit each other's knowledge, beliefs, and preferences. Moreover, a consensus can be reached in a coordinated way, when facing project challenges. Also, it would be easy for them to switch alternatively between implicit coordination and communication because the interaction partners know what they know/like and they do not know/like. Information and clarification would be offered or asked for at the right moment. The following comments from P1-PM nicely substantiate the argument.

> *"If I know that P1-DEV is passionate about a topic, I can go more in-depth and share that knowledge. If I know that he doesn't really care about that topic, then I'll probably keep it at a very, very crisp executive summary because he may disregard the rest anyways."* (P1-PM, the high understanding and high similarity condition)

Besides implicit coordination, given accurate understanding, interaction partners are more likely to engage in intensive knowledge exchange and integration processes. Accurate understanding that facilitates the adaptive interaction goes beyond knowledge identification and retrieval suggested by Transactive Memory Systems (Wegner, 1995) and is more similar to previous research on expertise coordination (Faraj & Sproull, 2000) where recognizing the need for expertise and bringing expertise to bear are needed. However, the construct of expertise coordination only describes the process of expertise coordination and fails to specify what understanding is needed for the process. An understanding of mental models helps coordination partners see the needs and can dynamically adjust to integrate knowledge. A deeper level of understanding of knowledge, beliefs, and preferences, rather than a basic understanding of expertise, is essential to expertise coordination.

Another barrier that needs to be overcome is affect. Similarity breeds attraction (Byrne, 1971) while differences may engender bias (Allport, 1954). Our results suggest a need to understand social aspects and individual needs and constraints because such understanding often brings about interpersonal citizenship behaviors. Helping behaviors help grow liking and benevolence trust. A tricky part is that understanding alone may not be sufficient for interpersonal citizenship behaviors. There needs to be benevolence trust to a certain degree. In a workplace, benevolence trust is highly related to competence trust (McAllister, 1995). Our results support the link between benevolence trust and competence trust.

Also, in order to build benevolence trust, a good starting point is to understand others' expertise because an understanding of expertise is a source of competence trust.

4.6.3 Research Question 3: How do Organizations Cultivate Understanding?

Considering benefits of accurate understanding , our cases suggest three ways to enhance accurate understanding.

First, frequent interaction always helps understanding but some guidance may be needed. Some people may not be aware of the benefits of understanding and choose to focus on their work. Guided reflexivity activity (Gurtner, Tschan, Semmer, & Nägele, 2007) (e.g., retrospective meetings), in which project members reflect upon group tasks collectively and individually, along with project manager's intervention on building cross-understanding (e.g., frequent check-in) should be beneficial.

Second, mentoring relationships provide a strong bond between project managers and developers and therefore build up understanding. Consistent with previous research, informal mentoring relationships have demonstrated strong effects on mentoring outcomes (Ragins & Cotton, 1999). However, our cases only suggest that mentoring should be useful for mentees who have not developed their mental models (i.e., junior co-workers). For people who have more stable mental models (i.e., senior co-workers), it should bear in mind that mentoring is not designated to change others' mental models. Rather, it may be more beneficial to help senior co-workers understand other project members' mental models.

Lastly, anchoring events change co-workers' understanding of each other. Major events may differ from person to person. Organizations should create extreme events for project members to instill memory episodes that can be used to face different project challenges. Literature on group learning in crisis may be useful. For instance, through simulated project crisis cases, project managers and developers may learn more from each other about how they would deal with the situation (Borodzicz & Van Haperen, 2002).

4.6.4 Limitations

First, a lack of cases in the low understanding and low similarity condition may raise questions of theoretical replication, although the problem may be ameliorated to a certain degree as the results have already shown two cases in the high similarity and low understanding condition have less effective coordination. It is less likely that the extreme cases in the low understanding and low similarity condition would have more effective work relationships than those two cases. Furthermore, the anecdote between P5-PM and P6-DEV (this is not a formal case but can be categorized as the low similarity and low understanding condition) described previously in discussion provides initial support. Nonetheless, extreme cases should provide more vivid stories about the impacts of a lack of understanding and similarity. There is a need to continue searching for such cases. To overcome challenges of identifying such extreme cases, one potential source is to identify newly-hired employees who need to work closely with their co-workers (either project managers or developers). In this case, resistance to participation should be weaker as their challenging work relationships are due to the time together rather than long-lasting interpersonal incompatibility.

Second, the sample from the high task-interdependent condition raises questions of generalizability. It is assumed that mental models are needed when there is a need to manage the interdependencies of tasks. However, in business settings, work relationships between project managers and developers are not limited to the direct interaction between each other. For instance, project managers have considerable latitude to influence project coordination mechanisms. Project members may judge project managers' competence trust based on these coordination mechanisms. Therefore, even though project managers do not work closely with project members, their understanding of the project members would be helpful for the formulation of more appropriate coordination mechanisms. Future research testing the theoretical model in the low task-interdependence context may add new insight into the model.

Third, our results indicate a diminishing role of similarity. However, two cases in the low similarity condition may not represent extremely dissimilar mental models. Although their mental models demonstrate high differences in the content and structure, project managers and developers in these two cases share overlapping beliefs, probably due to influences of organizational culture. It would be worthwhile to examine whether highly accurate understanding can mitigate dissimilarity in terms of both knowledge and beliefs. This is more likely to exist in the context of newly merged organizations or project teams composed of team members from different organizations or countries. Future research may find it more challenging to develop high cross-understanding in the context, although I argue that the effectiveness of three antecedents: mentorship, anchoring events, and frequent interaction, should still hold.

Fourth, for the assessment of work relationships, I assess participants' self-reported attitudes toward their co-workers, the interaction behaviors (coordination and interpersonal citizenship behaviors), and perceived work relationships. Due to social desirability bias, it is possible that people are reluctant to mention negative aspects of work relationships. The issue is of less concern in this research because participants who have above average work relationships genuinely showed their feelings of work relationships; for instance, P7-PM responded to the perceived work relationship questions as below:

> *"I would say above average, honestly. Like, P7-DEV and I get along on a personal level and a professional level. He trusts me, and I trust him that he can do pretty much anything we can ask him to do. So we get along very well in that regard."* (P7-PM in the high similarity and high understanding condition)

Two pairs who reported the average work relationship (P2 and P10 in the high similarity and low understanding condition), they may suffer social desirability bias and hide some negative aspects of work relationships (i.e., the report of findings is conservative). However, this only strengthens the argument that understanding does matter in work relationships. Having said that, I agree that multiple sources of qualitative evidence, such as direct observation and questionnaire, should enhance "trustworthiness and authenticity" (Patton, 2002) of the findings.

Fifth, the computational representation of mental models (i.e., boxes, links, and their relationships are akin to computer operations in terms of input, process, and output) is the primary lens used in this dissertation to understand both project managers and developers' knowledge and beliefs about ISDP. This lens captures knowledge and beliefs that people apply to the general context (i.e., global mental models). However, scholars who believe in the fuzzy and fluid nature of mental models may suggest looking into the process of sense-making where people scan the environment, make an interpretation, and act upon meanings derived from sense-making (Gioia & Chittipeddi, 1991). During the process, individuals search for situational meanings and form situational mental models. When there are discrepancies between global mental models and situational mental models, people experience high level of distress as people may strive to resolve discrepancies and even need to violate their belief systems to accommodate situational challenges (Park, 2010). Work relationships may falter because both project managers and developers may not able to predict what others would behave. This uncertainty is particularly serious when people are under high pressure to revolve equivocal tasks.

In this study, the situational mental model may play a less significant role because extremely stressful events - one of the prerequisite conditions of situational mental model (Weick, 1988) – did not exist in my six cases. Nevertheless, future research can take both global and situational mental models into consideration and examine how they play out during collaboration and coordination. This would add a boundary condition to the current theoretical model – how and when global mental models (i.e., ISDP mental models) matter.

Lastly, I follow Dubé & Paré's (2003) recommendation for rigor positivist case study and assess the extent to which this study has achieved rigor (see Table 18).

Table 18 Assessment of Rigor of Positivist Case Study

Criteria	Assessment of Study 2
Research Design	
Clear research questions	Yes, I stated clear research questions explicitly. Research Questions: How does the interplay of ISDP mental models affect work relationships between project managers and developers? (1) How does the similarity of ISDP mental models affect work relationships? (2) How does an understanding of others' ISDP mental models affect work relationships? (3) What are the major drivers of an accurate understanding and misunderstanding of others' ISDP mental models?
A priori specification of constructs and clean theoretical slate	Yes, I clearly define the ISDP mental model construct and cross-understanding. Multiple theories have discussed its theoretical meaning and importance in the work relationships. For work relationships, I specify the general notion of the concept and introduce multiple dimensions.
Multiple-case design	Yes, six cases were used to develop a theory.
Replication logic in multiple-case design	Yes, I follow Yin's (2009) theoretical replication and literal replication.
Unit of analysis	Yes, the unit of analysis is work relationships between a project manager and developer.
Pilot case	No. However, I conducted cases sequentially. The first case I picked is in the high cross-understanding condition, which helped understand the novel construct.
Context of the study	Yes, I describe the background of organizations (e.g., products, services, and methodologies used) where the dyads work. I also provide information about characteristics of the dyads that may be related to work relationships (e.g., working history).
Team-based research and different roles for multiple investigators	No, given the nature of dissertation (training for independent research), I did not work with a team of researchers in data collection and analysis. For future publication, there is a need to recruit coders who are blind to the theoretical framework to recode the interview documents.
Data Collection	
Elucidation of the data collection process	Yes, I describe the number of interviews conducted, profile of interviewees, sampling strategy, and the data collection procedures in Section 4.4.4.
Multiple data collection methods and mix of qualitative and quantitative data	Yes, a multi-method approach is used, including data collected from the survey and interviews.
Data triangulation	Yes, I combine interview data and survey data to validate the similarity of mental models and accurate understanding. With regard to work relationships, I assess participants' self-reported attitudes toward their co-workers, the interaction behaviors (coordination and interpersonal citizenship behaviors), and perceived work relationships. The accounts of work relationships from the dyad avoid common source bias.
Case study protocol and	Yes, I define the interview protocol and use MAXQSA as a case study database.

case study database	
Data Analysis	
Elucidation of the data analysis process	Yes, I explain the data analysis process in Section 4.4.5.
Field notes, coding, data displays, and flexible process	Yes (1) I write brief field notes for recording verbal and nonverbal cues in the context of communication. (2) Systematic coding with coding hierarchies (master codes and sub-codes) is adopted to reduce complexity of data. Moreover, during the process of analysis procedure, I rely on network display (MAXMap in the MAXQDA software) to map the associations among master codes and sub-codes, which helps crystallize my thoughts as I spend a couple of months immersing myself in data. Furthermore, I use data matrix to show similarity and contrast of results between different conditions. This conveys synthesized ideas to readers. (3) I take a neutral stance in interviewing and data analysis without judgment by showing openness to potential responses and explanations. The attitude rests upon a belief in the ultimate value of what analysis will yield.
Logical chain of Evidence	Based on Dubé & Paré's (2003) criterion[26], I present a strong logical chain of evidence. I introduce the cases in conditions (high/low similarity and understanding), describe concepts, compare conditions, establish claims, and uses evidence to support claims.
Cross-case patterns	Yes, I show within-condition similarities along with inter-condition differences.
Quotes	Yes, I use quotes to present compelling evidences and thick descriptions.
Project reviews	No. I did not ask participants to assess the credibility of interpretations.
Comparison with extant literature	Yes, I compare the findings with literature from team cognition and work relationships.

[26] "In order to assess whether the authors of a case report had maintained a chain of evidence, we evaluated the extent to which we were able to move from one portion of the case study to another, with minimal cross-referencing to methodological procedures and to the resulting evidence." (p. 618)

Chapter 5

Contributions and Future Research

Managing ISD projects in organizations is one of the most difficult aspects of ISD. Project members have to deal with not only organizational and technical complexity but also dynamic complexity coming from changes (Xia & Lee, 2003). To respond to the complexity of ISD projects, project members play a critical role in the process of managing ISD projects. More importantly, how they collaborate has significant effects on the capacity of a project team for project complexity. To this end, this dissertation shifts the focus away from ISD methodologies and places it on the critical role of people. I maintain that it is project members and the interpersonal dynamics among project members that contribute to the successful outcome of a project. Although the people component has rarely been overlooked in prior literature, it is lacking is a fresh perspective on why project members respond to project complexity differently and how project members collaborate when they have different knowledge and beliefs. A better understanding of project members' knowledge and beliefs can address the central questions in the study of ISD project management. This dissertation adopts the cognitive perspective with a mental mode approach, which provides a sophisticated lens to represent project members' mindsets and to prescribe their information processing, decision-making, and behaviors.

Following this perspective, in study 1 (Chapter 3), I defined a new construct called ISDP mental models and posed a set of research questions: W*hat is a project member's ISDP mental model? What are the major ISDP concepts that comprise a project member's knowledge and belief structures about the management of ISD projects and how are these concepts interrelated?* Findings of study 1, based on the content analysis of rich literature on ISD with the elicitation of mental models from ISD experts, yield a list of forty ISD concepts. They represent dominant concepts that ISD professionals possess. Study 1 further addressed why mental models are organized differently. ISD professionals' mental models are shaped by four types of evaluative beliefs (i.e., team, customer, enterprise, and product) on the basis of two dimensions (i.e., "concerns about design vs. people" and "concerns about open vs. closed project environment").

Building upon the basis of study 1, study 2 (Chapter 4) strived to improve our understanding about the impact of project members' mental models on ISD projects. I chose work relationships between project managers and developers to understand the interplay of ISDP mental models not only because the dyadic relationship is a suitable level of analysis to unfold complex interaction of mental models but also because the dyadic relationship between project managers and developers are key to the success of a project. By focusing on the influences of ISDP mental models, study 2 explored potential answers for three research questions via multiple case studies: First, *how does the similarity of ISDP mental models affect work relationships?* Second, *how does an understanding of others' ISDP mental models affect work relationships?* Third, *what are the major drivers of an accurate understanding and misunderstanding of others' ISDP mental models?* The findings provide evidence that the similarity of mental models is not a determinant factor for work relationships. Rather, an accurate understanding of ISDP mental models has an effect on how project managers and developers manage the interdependencies of tasks and people. Accordingly, the dyad is able to anticipate the intention and needs of one another, culminating implicit coordination and knowledge integration.

This dissertation synthesizes the body of ISD project knowledge with strong support of theories and empirical evidence. I hope that this discussion of ISDP mental models and its application in ISD project management creates research interests in further exploration of cognitive factors in ISD project management. I would like to close this discussion by discussing theoretical and practical contributions, research challenges, and future research direction.

5.1 Theoretical Contributions

5.1.1 Building Research from the New Construct

Construct is "conceptual abstractions of phenomena that cannot be directly observed" (MacCorquodale & Meehl, 1948) and is the foundation of theory development (Bacharach, 1989). Prior research on ISD project management has borrowed constructs from other disciplines, such as psychological contract (Koh, Ang, & Straub, 2004), escalation of commitment (M. Keil, Mann, & Rai, 2000), and control (Kirsch, 1997), to understand the intricacy of ISD project management. Different from prior literature, this dissertation identifies and develops a new construct native to the IS discipline – ISDP mental models - to comprehend how ISD professionals construe work practices pertaining to ISD projects and how concepts are formed and organized in ISD professionals' minds. Furthermore, this dissertation adopts a computational and quantitative approach, different from a constructionist view of cognitive processing (e.g., Vlaar, van Fenema, & Tiwari, 2008), to assess how ISD professionals make sense of the management of ISD, which is likely to affect critical decisions in response to external and internal demands. The new construct pushes the notion of mental model from a metaphor toward a construct with greater clarity and more precise measurement.

Besides a clear conceptualization, the ISDP mental model construct provides an operationalization of underlying concepts and their linkages, which some other constructs may not provide. For instance, regarding conceptual abstractions of phenomena, the construct of control aims to understand a mechanism used by a controller to discourage a controllee's anticipated opportunistic behaviors (Ouchi & Maguire, 1975) whereas the construct of ISDP mental models is to uncover ISD professionals' knowledge and belief structures that help them understand, conduct, and manage ISD projects. However, when looking into the operationalization of construct, there are different implications. Take the control construct for example, multiple types of mechanisms, such as behavior control, clan control, and outcome control, have been identified; each type requires different informational and social requirements to be effective (Kirsch, 1997). Different mechanisms of the control construct all could interest researchers and practitioner to study the choice of control mechanisms in ISD projects (Kirsch, 1997; Tiwana & Keil, 2009). Nonetheless, specific control concepts and practices are scattered across different research studies. Conversely, the ISDP mental model construct provides ISDP concepts that practitioners apply in ISD projects along with a summarization of specific practices (see Appendix A). Not only can researchers and practitioners know 40 fundamental ISDP concepts but also know how these concepts are related. This dissertation responds to Nelson et al.'s (2000) call for "evocative research" on understanding the phenomenon at the in-depth level by studying how practitioners operationalize underlying concepts of a construct. By synthesizing numerous principles from the long-standing literature on software engineering and project management and by discovering how practitioners operationalize the general theory of ISD project knowledge, I believe that the results are of value for researchers to better understand ISD project management and for practitioners to reflect upon the accumulated knowledge. Furthermore, the advanced understanding of the ISDP construct answers the lament of a lack of theoretical research in project management.

Lastly, the construct discovered four evaluative beliefs. The findings update the three evaluative beliefs suggested by Kumar & Bjorn-Andersen (1990). The original socio-political belief is replaced by customer-oriented belief and team-oriented belief. The shift can be regarded as the ethos of existing ISD project environment, which requires more close team collaboration and delivery of value to customers in order to respond to changes. Furthermore, these four beliefs have solid content (as shown in Figure 12 to Figure 15), which provides both researchers and practitioners an opportunity to understand the association between concepts and beliefs. Accordingly, they have better knowledge about why ISDP professionals evaluate certain development principles and practices in a positive light and accordingly follow them.

5.1.2 Clarification of Sharedness

The understanding of ISDP mental models is the core of contribution to shared cognition research by providing better knowledge about what "sharedness" means (Cannon-Bowers & Salas, 2001). The concept of sharedness in cognition is comprised of multiple notions, such as the overlapping of mental models (i.e., people need to have mutual knowledge to a certain degree), the similarity of mental models (e.g., team members need to have highly similar knowledge about taskwork and teamwork), and the distributed mental models (e.g., knowledge should be appropriately distributed across team members). The theoretical model of study 2 suggests that an understanding of mental models should be at the very heart of shared cognition for collaboration. It brings a distinctive benefit to implicit coordination and knowledge integration. Although prior studies suggest that the similarity of mental models facilitates implicit coordination (Espinosa et al., 2004) and knowledge exchange and integration (Marks, Sabella, Burke, & Zaccaro, 2002), the findings of study 2 suggest that the impact of similarity is weakened, if not removed, when an understanding of mental model is low. I posit that an understanding of mental models deserves a place in shared cognition not only for its significant influences to interpersonal dynamics but also for construct clarity.

There is also a need for further clarification regarding the relationships between different notions of sharedness. The literature emphasized the end state of shared cognition, that is, what impacts shared cognition, with a specific focus on accuracy of individual mental models and similarity of mental models among team members, can bring to team processes and performance (see recent review in Mohammed et al., 2010). Thus, it tends to overlook the process of interaction in obtaining different types of sharedness. As evidence shown in Section 4.6.1, an understanding of mental models can be a precondition of shared cognition in terms of overlapping, similarity, and distribution. This line of inquiry can fill the gap in the existing literature.

5.1.3 On Trust: Is Trust the Panacea for ISD Project Management?

Research interests in trust have grown dramatically over a decade. The accumulated evidence supports that trust improves individual task performance, increases citizenship behaviors, and reduces counterproductive behaviors (Colquitt et al., 2007). Trust also helps team performance because it provides the climate of psychology safety and allows team members to exchange knowledge and ideas with fewer concerns of vulnerability (Edmondson, 1999). That is to say, when people have "faith in the trustworthy intentions of others" (benevolence trust) and "confidence in the ability of others" (competence trust), there may be more intensive knowledge exchange (Politis, 2003).

Although research on trust has provided the myriad and often subtle benefits such trust entails, this dissertation suggests understanding may provide benefits that trust cannot offer. The results of study 2 showed some preliminary evidence that, in the high benevolence and competence trust environment, the coordination pairs who possess accurate understanding of one another's ISDP mental models demonstrated implicit coordination and effective knowledge integration. Understanding allows those pairs to frame problems in a similar fashion, anticipate others' needs and preferences, and provide information in a timely manner. In other words, while trust can be panacea, understanding can be "personalized gene-targeted" therapy, customized for different project members and for different project domains. They should be more effective in contributing to better solutions. The preliminary findings in this research are based on several cases with reasonably rich information. To substantiate this argument, I suggest a larger scale survey for the examination of working pairs who have high trust but different levels of understanding.

5.2 Practical Contributions

What can we learn from ISDP mental models. For the professional community, the ISDP mental model construct with the assessment procedures offers educational benefits. People are generally unaware of their mental models until they are challenged, or until they experience new concepts and work practices, or until the mental model is made overt and explicit through a theoretical framework. The ISDP mental models identified in this dissertation provide ISD professionals with a frame of reference about what they can do and help them to reflect upon their application of different concepts. Specifically, people can look into beliefs that they may have overlooked and paid less attention to. For senior ISD professionals, it is important to step out of comfort zones and consider other project members' stand. Thus, when different opinions on ways to address project issues emerge, project members are able to attend to issues rather than attribute other project members' ideas to their lack of knowledge. For junior ISD professionals, the ISDP mental models help them avoid a trap of tunnel views and expand their repertoires of knowledge for dealing with the complexities of projects. The underlying associations of concepts provide guidance on what concepts should be understood and implemented as a whole to enhance development effectiveness (e.g., simplicity of design complementary to continuous integration (Highsmith, 2002; Nerur, Mahapatra, & Mangalaraj, 2005)). To achieve the above goals, this dissertation develops an assessment tool. Managers or ISD professionals can use it not only to monitor their own mental models but also to track the changes of mental models. Furthermore, the assessment tool could be used to help understand their co-workers' or subordinates' mental models (with the permission from their co-workers and subordinates). This could provide ISD professionals and organizations with a useful devise for training and team management.

When birds of a feather do not flock together. Organizations have been searching for ways to harness diverse expertise in ISD project teams while reducing process losses due to differences. This dissertation sheds light on the diversity issues. The theoretical model of study 2 indicates that different interpersonal outcomes entail two types of understanding: task-aspect and personal aspect understanding. With regard to task-aspect understanding, this dissertation discovers the importance of understanding of ISDP mental models and its multiple influences on work relationships. The understanding of ISDP mental models moves previous discussion of task-aspect understanding to a new territory. It is suggested that an understanding of expertise in general is not sufficient for the management of tasks and people; there is a need for project members to understand domain-specific concepts that others possess. With regard to the personal aspect understanding, the results support the importance of social understanding and understanding of individual needs and constraints. The personal aspect understanding increases benevolence trust and interpersonal citizenship behaviors. Based on various benefits of the typology of understanding, this dissertation offers a suggestion on the management of diverse expertise. Organizations can work on different interventions for the enhancement of different types of understanding. For instance, Chiu and Staples (2013) showed the interaction between personal-aspect understanding via blogging and interpersonal attraction mitigates the intergroup bias in dispersed team. The intervention using social media can be regarded as the frequent interaction antecedent indicated in the theoretical model of study 2. I suggest that organizations should be creative to design work environment that enhances understanding.

5.3 Limitations

In Section 3.5.3 and Section 4.6.4, I have specified limitations for study 1 and study 2. In this section, I would discuss general challenges the research program may face. Mental model is "a simplified representation of an information world" and is useful only when it can accurately represent the specified information world (Walsh, 1995). Although research evidence suggests that task-specific mental models can be mapped and be used to predict behaviors and performance (See literature review in Section 2.3 and Section 2.4), the challenges for this dissertation are whether the world of ISD project management is too complex to capture and how a simplified representation can predict behaviors of project members. Findings from this dissertation may address the first challenge. Via a series of inquiry and elicitation, I believe that the derived ISDP mental model finds a reasonable balance between simplicity and comprehensiveness (see Section 3.4.1). It demonstrates an essential set of concepts and beliefs that ISDP professionals possess. The accuracy of derived mental models in terms of knowledge and belief is confirmed by practitioners (see Section 3.5.2). In other words, the ISDP mental model developed in this study can be used to approximate project members' mental models as closely as possible.

The answer to the second challenge is less gratifying. This dissertation provides evidence that maps an association between ISDP concepts and behaviors. For instance, in study 2, P3-

DEV's account for his planning knowledge and behaviors matches the "iterative and incremental development" concept in his mental model.

> *"..any given feature, I'll look at it, and I'll plan roughly how I want to approach solving the problem… I'm not necessarily thinking about every small piece of the solution right when I look at it…I'm not the type of developer who wants to go and spend a day looking at a task and breaking it up into: these are the 20 steps that I need to do to solve this because I know that once I get to step 5, I'm going to realize that, oh, wait, that really isn't the best way to do this. I actually should be doing it this way, and that's going to have an impact on all these other 15 steps that I already planned out. So I would rather look at it and say, "Okay, here are roughly the five things that I need to get done" and then get started on it."* (P3-DEV)

However, there is a less obvious link between the ISDP mental model as a whole and behaviors in response to complex project issues. That is, what are influences of nexus of concepts rather than a single concept on decision-making and actions? For instance, how would people whose mental models contain a web of concepts, such as iterative and incremental development and continuous integration, respond to time to market issues? Will they behave differently from people whose mental models contain a web of other concepts, such as plan-driven development and close control over software development processes? It seems that people act in accordance to their mental models as study 2 implied (if they do not, an accurate understanding of others' mental models would be ineffective in predicting implicit coordination). Nevertheless, the question remains: when and how these concepts are activated for complex project issues? It is important to spell out the contextual conditions under which a nexus of concepts will be triggered because it strengthens the prescriptive power of the construct. One potential approach to addressing this issue is to identify critical incidents in projects and analyze decision-making processes of the participating members. We can analyze the set of concepts that has been used and find patterns across critical incidents that share similar problem characteristics. Furthermore, it would be worthwhile to identify factors, such as emotion, that may confound the uses of ISDP mental models.

5.4 Future Research

The ISDP mental model construct has unique value to ISD professionals and communities. Below, I offer some more potential research opportunities.

At the individual level, future research can look into ideal mental models for different types of ISD projects (the theme of mental model accuracy). For instance, in offshoring projects, a project team may need to enact a portfolio of concepts and practices that can address communication challenges due to dispersed locations and knowledge gaps between vendors and clients. Concepts from the team-oriented belief and customer-oriented belief may be beneficial to maintain coordination efficiency, trust, and transparency. Once the ideal mental model for the specific type of projects is developed, it would be worthwhile to study potential training and managerial interventions on how to develop and/or change ISD professionals' mental models.

At the interpersonal and team level, this dissertation studied work relationships between project managers and developers and suggested an accurate understanding of mental models as a potential aid for diversity issues. However, considering the prevalence of offshoring projects across countries, it is important to consider the role of understanding in a more severely diversity context. When diversity of mental models and other demographic characteristics align together, such as gender, ethnicity, and nationality, stronger intergroup bias would be created according to faultline theory (D. C. Lau & Murnighan, 1998). For instance, if a project team of four members include a white female project manager and a white female system analyst in their 40's and two Asian male programmers in their 20's, the two subgroups in this team have high within group similarity in gender, age, and ethnicity. If it happens that the subgroup members also share similar

ISDP mental models, these four characteristics would create strong faultlines. The prevalence of global collaboration makes such extreme diversity common. I contend that the understanding of ISDP mental models should be more critical to ensure coordination effectiveness but it should be more challenging to achieve high cross-understanding.

At the organizational level, I suggest that future research can direct attention to the fit between project members' ISDP mental models and organizational ISD project management mechanisms in order to decrease project members' intention to quit, low job satisfaction, and low commitment (Kristof- Brown, Zimmerman, & Johnson, 2005). It should be noted that the fit is not about removal of mechanism that project members do no prefer. Instead, it should integrate diverse knowledge and beliefs that project members possess, which in turn is likely to create an optimal collective ISDP mental model for a project environment. The assessment procedures developed in study 1 can help organizations specify the origins of difficulties and conflicts around ISD. The results can serve as the common ground for communication, which would reduce misunderstanding and increase the accuracy of understanding. The exercise can serve as an anchoring event suggested by the theoretical model of study 2. Consequently, the more compatible and effective organizational ISDP management mechanisms are developed to improve collaboration in ISD projects. Very little is known about the process to attain the fit between project members and organizations, and I suggest that action research is an appropriate mean to understand the phenomenon.

Lastly, ISDP management blends best practices from traditional project management and software engineering. The accumulated knowledge continues to grow into a variety of approaches. Practitioners from other disciplines have been seeking advice from our discipline in order to manage projects in effective ways; for instance, recent successes of Agile in ISDPs has drawn a substantial amount of interest from non-ISD practitioners (Highsmith, 2009). However, we have limited knowledge about whether all ISDP concepts are applicable to other types of projects. I suggest that researchers compare ISDP mental models with mental models of team members working in other types of projects. The differences may reveal unique knowledge held by the IS discipline. Based on the result, we can further investigate how the knowledge can contribute to other disciplines. Furthermore, it is possible that some concepts from other disciplines can be introduced to ISDPs. This cross-pollination could enhance the maturity of project management in both IS and non-IS areas.

5.5 Conclusion

In sum, this dissertation sheds new light on ISD project management and shared cognition research. This dissertation develops and defines mental models about ISD project knowledge in a rigorous way. One potential impact of ISDP mental models on project management has been demonstrated by showing how the interplay of mental models between project managers and developers affect work relationships. The research perspective and approach employed here would seem to be useful in future research on ISD project management in particular and on interpersonal dynamics in general. The utility lies in its access to knowledge and beliefs ISD project members possess. This dissertation unveils only part of how complex human minds operate in the complex ISD project environment. There are abundant opportunities for practitioners and researchers to advance our understanding of human minds in ISD projects. I hope to see more diverse and rich discussion on the subject in the near future.

References

Abelson, R. P. (1979). Differences between belief and knowledge systems. *Cognitive Science*, *3*(4), 355–366.

Abrahamsson, P., Oza, N., & Siponen, M. T. (2010). Agile software development methods: A comparative review. In T. Dingsøyr, T. Dybå, & N. B. Moe (Eds.), *Agile software development* (Vol. 1, pp. 31–59). New York, NY: Springer.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: A comparative analysis. *Proceedings of the 25th International Conference on Software Engineering*, 244–254.

Ackermann, F., & Eden, C. (2005). Using causal mapping with group support systems to elicit an understanding of failure in complex projects: some implications for organizational research. *Group Decision and Negotiation*, *14*(5), 355–376.

Adman, P., & Warren, L. (2000). Participatory sociotechnical design of organizations and information systems–an adaptation of ETHICS methodology. *Journal of Information Technology*, *15*(1), 39–51.

Agarwal, R., & Ferratt, T. W. (2002). Enduring practices for managing IT professionals. *Communications of the ACM*, *45*(9), 73–79. doi:10.1145/567498.567502

Allen, V. L., & Wilder, D. A. (1975). Categorization, belief similarity, and intergroup discrimination. *Journal of Personality and Social Psychology*, *32*(6), 971–977. doi:10.1037/0022-3514.32.6.971

Allinson, C. W., Armstrong, S. J., & Hayes, J. (2001). The effects of cognitive style on leader-member exchange: A study of manager-subordinate dyads. *Journal of Occupational and Organizational Psychology*, *74*, 201–220.

Allport, G. (1954). *The nature of prejudice*. Boston, MA: Addison-Wesley.

Ambler, S., & Vizdos, M. (2009). *Agile practices survey 2009*. Retrieved from http://www.ambysoft.com/surveys/practices2009.html

Andersen, N., Kensing, F., Lundin, J., Mathiassen, L., Munk-Madsen, A., Rasbech, M., & Sorgaard, P. (1990). *Professional systems development: Experience, ideas and action*. Hemel Hempstead, UK: Prentice Hall.

Argyris, C. (1982). *Reasoning, learning, and action: Individual and organizational*. San Francisco, CA: Jossey-Bass.

Armstrong, C. P., & Sambamurthy, V. (1999). Information technology assimilation in firms: The influence of senior leadership and IT infrastructures. *Information Systems Research*, *10*(4), 304–327.

Armstrong, D. J. (2005). Causal Mapping: A Discussion and Demonstration. In V. Narayanan & D. J. Armstrong (Eds.), *Causal mapping for research in information technology* (pp. 20–45). Hershey, PA: Idea Group Publishing.

Armstrong, D. J., & Hardgrave, B. C. (2007). Understanding mindshift learning: The transition to object-oriented development. *MIS Quarterly*, *31*(3), 453–474.

Armstrong, D. J., & Narayanan, V. K. (2005). Expanding Horizons: Juxtaposing Causal Mapping and Survey Techniques. In V. K. Narayanan & D. J. Armstrong (Eds.), (pp. 174–194). Hershey, PA: Idea Group Publishing.

Ashkanasy, N. M., & O'Connor, C. (1997). Value congruence in leader-member exchange. *The Journal of Social Psychology*, *137*(5), 647–662.

Ashmos, D. P., & Huber, G. P. (1987). The systems paradigm in organization theory: Correcting the record and suggesting the future. *Academy of Management Review*, *12*(4), 607–621.
Aveling, B. (2004). XP lite considered harmful? In *eXtreme Programming and Agile processes in software engineering* (pp. 94–103). New York, NY: Springer.
Avison, D. E., & Fitzgerald, G. (2003). Where now for development methodologies? *Communications of the ACM*, *46*(1), 78–82.
Avison, D. E., & Fitzgerald, G. (2007). *Information systems development: Methodologies, techniques and tools* (4th ed.). New York, NY: McGraw-Hill.
Avison, D. E., & Wood-Harper, A. T. (1990). *Multiview: An exploration in information systems development*. Oxford, UK: Blackwell Scientific.
Avison, D. E., Wood-Harper, A. T., Vidgen, R. T., & Wood, J. R. G. (1998). A further exploration into information systems development: The evolution of Multiview2. *Information Technology & People*, *11*(2), 124–139.
Bacharach, S. B. (1989). Organizational theories: Some criteria for evaluation. *Academy of Management Review*, *14*(4), 496–515.
Baines, T., Lightfoot, H., Williams, G. M., & Greenough, R. (2006). State-of-the-art in lean design engineering: A literature review on white collar lean. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *220*(9), 1539–1547.
Ballinger, G. A., & Rockmann, K. W. (2010). Chutes versus ladders: Anchoring events and a punctuated-equilibrium perspective on social exchange relationships. *Academy of Management Review*, *35*(3), 373–391.
Barki, H., & Hartwick, J. (2001). Interpersonal conflict and its management in information system development. *MIS Quarterly*, *25*(2), 195–228.
Barki, H., Rivard, S., & Talbot, J. (2001). An integrative contingency model of software project risk management. *Journal of Management Information Systems*, *17*, 37–70.
Barsalou, L. W. (1991). Deriving categories to achieve goals. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 27, pp. 1–64). San Diego, CA: Academic Press.
Bassellier, G., & Benbasat, I. (2004). Business competence of information technology professionals: Conceptual development and influence on IT-business partnerships. *MIS Quarterly*, *28*(4), 673–694.
Bauer, T. N., & Green, S. G. (1996). Development of leader-member exchange: A longitudinal test. *Academy of Management Journal*, *39*, 1538–1567.
Bazerman, M. H., Curhan, J. R., Moore, D. A., & Valley, K. L. (2000). Negotiation. *Annual Review of Psychology*, *51*(1), 279–314.
Beck, K. (2000). *Extreme programming explained: embrace change*. Boston, MA: Addison-Wesley.
Benbasat, I., & Zmud, R. W. (1999). Empirical research in information systems: The practice of relevance. *MIS quarterly*, *23*(1), 3–16.
Bharadwaj, A. S. (2000). A resource-based perspective on information technology capability and firm performance: An empirical investigation. *MIS Quarterly*, *24*(1), 169–196.
Blau, P. M. (1964). *Exchange and power in social life*. New Brunswick, NJ: Transaction Publishers.
Blickensderfer, E., Cannon-Bowers, J. A., & Salas, E. (1998). Cross-training and team performance. In J. A. Cannon-Bowers & E. Salas (Eds.), *Making decisions under stress:*

Implications for individual and team training (pp. 299–311). Washington, DC: American Psychological Association.

Boehm, B., & Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*. Boston, MA: Addison-Wesley Professional.

Booch, G., Maksimchuk, R. A., Engle, M. W., Conallen, J., Houston, K. A., & Young, B. J. (2007). *Object-Oriented analysis and design with applications* (3rd ed.). Boston, MA: Addison-Wesley Professional.

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. New York, NY: Springer.

Borg, I., Groenen, P. J., & Mair, P. (2013). *Multidimensional Scaling*. New York, NY: Springer.

Borgatti, S. P. (2013). *NetDraw software for network visualization*. Lexington, KY: Analytic Technologies.

Borodzicz, E., & Van Haperen, K. (2002). Individual and group learning in crisis simulations. *Journal of Contingencies and Crisis Management*, *10*(3), 139–147.

Bower, G. H., & Hilgard, E. R. (1981). *Theories of learning*. Englewood Cliffs, NJ: Prentice Hall.

Browne, G. J., & Ramesh, V. (2002). Improving information requirements determination: A cognitive perspective. *Information & Management*, *39*(8), 625–645.

Burke, C. S., Stagl, K. C., Salas, E., Pierce, L., & Kendall, D. (2006). Understanding team adaptation: A conceptual analysis and model. *Journal of Applied Psychology*, *91*(6), 1189–1207.

Butler, T., & Murphy, C. (2008). An exploratory study on IS capabilities and assets in a small-to-medium software enterprise. *Journal of Information Technology*, *23*, 330–344.

Byrd, T. A., Cossick, K. L., & Zmud, R. W. (1992). A synthesis of research on requirements analysis and knowledge acquisition techniques. *MIS Quarterly*, *16*(1), 117–138.

Byrne, D. E. (1971). *The attraction paradigm*. New York, NY: Academic Press.

Cannon-Bowers, J. A. (2007). Fostering mental model convergence through training. *Research in Multi Level Issues*, *6*, 149–157.

Cannon-Bowers, J. A., & Salas, E. (2001). Reflections on shared cognition. *Journal of Organizational Behavior*, *22*(2), 195–202. doi:10.1002/job.82

Cannon-Bowers, J. A., Salas, E., & Converse, S. (1993). Shared mental models in expert team decision-making. In N. J. Castellan (Ed.), *Individual and group decision making* (pp. 221–246). Hillsdale, NJ: Lawrence Erlbaum Associates.

Case, P., & Piñeiro, E. (2009). Stop whining, start doing! Identity conflict in project managed software environments. *Ephemera*, *9*, 93–112.

Chakraborty, S., & Sarker, S. (2010). An exploration into the process of requirements elicitation: A grounded approach. *Journal of the Association for Information Systems*, *11*, 212–249.

Chan, D. (1998). Functional relations among constructs in the same content domain at different levels of analysis: A typology of composition models. *Journal of Applied Psychology*, *83*(2), 234–246. doi:10.1037/0021-9010.83.2.234

Chang, A., Chu, C.-C., Chi, S.-C., & Lo, H.-H. (2010). Users verses IT: Understanding prejudice in information systems development project teams. In *Academy of Management Annual Meeting Proceedings 2010*.

Chattopadhyay, P., Glick, W., Miller, C. C., & Huber, G. (1999). Determinants of executive beliefs: Comparing functional conditioning and social influence. *Strategic Management Journal*, *20*(8), 763–789.

Checkland, P., & Scholes, J. (1990). *Soft systems methodology in action*. New York, NY: John Wiley & Sons.

Chemuturi, M., & Cagley Jr, T. M. (2010). *Mastering software project management: Best practices, tools and techniques*. Fort Lauderdale, FL: J. Ross Publishing.

Chiaburu, D. S., & Harrison, D. A. (2008). Do peers make the place? Conceptual synthesis and meta-analysis of coworker effects on perceptions, attitudes, OCBs, and performance. *Journal of Applied Psychology*, *93*(5), 1082–1103.

Chiu, Y.-T., & Staples, D. S. (2013). Reducing faultlines in geographically dispersed teams: Self-disclosure and task elaboration. *Small Group Research*, (44), 498–531.

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, *81*(6), 961–971.

Clark, R. E., Feldon, D., van Merriënboer, J. J., Yates, K., & Early, S. (2008). Cognitive task analysis. In J. M. Spector, M. D. Merrill, J. J. G. van Merrienboer, & M. P. Driscoll (Eds.), *Handbook of research on educational communications and technology* (Vol. 3, pp. 577–593). New York, NY: Macmillan/Gale. Retrieved from http://www.learnlab.org/research/wiki/images/0/0b/Clarketal2007-CTAchapter.pdf

Cobb, C. G. (2011). *Making sense of Agile project management: Balancing control and agility*. Hoboken, NJ: Wiley.

Cockburn, A. (2005). *Crystal clear: A human-powered methodology for small teams*. Boston, MA: Addison-Wesley.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, *8*(2), 240–247.

Colquitt, J. A., LePine, J. A., Piccolo, R. F., Zapata, C. P., & Rich, B. L. (2012). Explaining the justice–performance relationship: Trust as exchange deepener or trust as uncertainty reducer? *Journal of Applied Psychology*, *97*(1), 1–15.

Colquitt, J. A., Scott, B. A., & LePine, J. A. (2007). Trust, trustworthiness, and trust propensity: A meta-analytic test of their unique relationships with risk taking and job performance. *Journal of Applied Psychology*, *92*(4), 909.

Cooke, N. J., Gorman, J. C., & Winner, J. L. (2007). Team cognition. In F. T. Durso (Ed.), *Handbook of applied cognition* (2nd ed., Vol. 2, pp. 239–268). Hoboken, NJ: Wiley.

Cooke, N. J., Salas, E., Cannon-Bowers, J. A., & Stout, R. J. (2000). Measuring team knowledge. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *42*(1), 151–173.

Cooke, N. J., & Schvaneveldt, R. W. (1988). Effects of computer programming experience on network representations of abstract programming concepts. *International Journal of Man-Machine Studies*, *29*, 407–427.

Cox, T. F., & Cox, M. A. (2000). *Multidimensional scaling*. London, UK: Chapman & Hall/CRC.

Coyle-Shapiro, J. A.-M., Shore, L. M., Taylor, M. S., & Tetrick, L. (2004). *The employment relationship: Examining psychological and contextual perspectives*. Oxford, UK: Oxford University Press.

Creswell, J. W. (2008). *Research design: Qualitative, quantitative, and mixed methods approaches*. London, UK: Sage Publications Ltd.

Cronin, M. A., Bezrukova, K., Weingart, L. R., & Tinsley, C. H. (2011). Subgroups within a team: The role of cognitive and affective integration. *Journal of Organizational Behavior*, *32*(6), 831–849.

Cronin, M. A., & Weingart, L. (2007). Representational gaps, information processing, and conflict in functionally diverse teams. *Academy of Management Review*, *32*(3), 761–773.

Cropanzano, R., & Mitchell, M. S. (2005). Social exchange theory: An interdisciplinary review. *Journal of Management*, *31*, 874–900. doi:10.1177/0149206305279602

Curtis, A., Wells, T., Higbee, T., & Lowry, P. (2008). An overview and tutorial of the Repertory Grid Technique in Information Systems research. *Communications of the Association for Information Systems*, *23*, 37–62.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, *31*, 1268–1287.

Dane, E. (2010). Reconsidering the trade-off between expertise and flexibility: A cognitive entrenchment perspective. *Academy of management review*, *35*, 579–603.

Davidson, E. J. (2002). Technology frames and framing: A socio-cognitive investigation of requirements determination. *MIS Quarterly*, *26*(4), 329–358.

Davidson, M. L. (1983). *Multidimensional scaling*. New York, NY: John Wiley & Sons.

Davis, M. H. (1983). Measuring individual differences in empathy: Evidence for a multidimensional approach. *Journal of Personality and Social Psychology*, *44*(1), 113–126. doi:10.1037/0022-3514.44.1.113

De Oliveira, S. B., Valle, R., & Mahler, C. F. (2010). A comparative analysis of CMMI software project management by Brazilian, Indian and Chinese companies. *Software Quality Journal*, *18*, 177–194.

DeChurch, L. A., & Mesmer-Magnus, J. R. (2010). Measuring shared team mental models: A meta-analysis. *Group Dynamics: Theory, Research, and Practice*, *14*(1), 1–14.

Deutsch, M. (1973). *The resolution of conflict: Constructive and destructive processes*. New Haven, CT: Yale University Press.

Dienesch, R. M., & Liden, R. C. (1986). Leader-member exchange model of leadership: A critique and further development. *Academy of Management Review*, *11*(3), 618–634.

Dirks, K. T., & Ferrin, D. L. (2002). Trust in leadership: Meta-analytic findings and implications for research and practice. *Journal of Applied Psychology*, *87*(4), 611–628.

Dubé, L., & Paré, G. (2003). Rigor in information systems positivist case research: current practices, trends, and recommendations. *MIS Quarterly*, *27*(4), 597–636. doi:10.2307/30036550

Dulebohn, J. H., Bommer, W. H., Liden, R. C., Brouer, R. L., & Ferris, G. R. (2012). A meta-analysis of antecedents and consequences of leader-member exchange: Integrating the past with an eye toward the future. *Journal of Management*, *38*(6), 1715–1759. doi:10.1177/0149206311415280

Dutta, S., Lee, M., & Van Wassenhove, L. (1999). Software engineering in Europe: A study of best practices. *Software*, *IEEE*, *16*, 82–90.

Edmondson, A. (1999). Psychological safety and learning behavior in work teams. *Administrative Science Quarterly*, *44*(2), 350–383.

Edwards, H. M., McDonald, S., & Young, S. M. (2009). The repertory grid technique: Its place in empirical software engineering research. *Information and Software Technology*, *51*, 785–798.

Edwards, J. R., & Cable, D. M. (2009). The value of value congruence. *Journal of Applied Psychology*, *94*, 654–677.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review*, *14*(4), 532–550.

Ellwart, T. (2011). Assessing coordination in human groups: Concepts and methods. In E. Boos, M. Kolbe, P. M. Kappeler, & T. Ellwart (Eds.), *Coordination in human and primate groups* (pp. 119–135). New York, NY: Springer.

Entin, E. E., & Serfaty, D. (1999). Adaptive team coordination. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *41*(2), 312–325.

Espinosa, J. A., Lerch, F. J., & Kraut, R. E. (2004). Explicit versus implicit coordination mechanisms and task dependencies: One size does not fit all. In E. Salas & S. Fiore (Eds.), (pp. 107–129). Washington, DC: American Psychological Association.

Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D. (2007). Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems*, *24*, 135–169.

Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. *Management Science*, *46*(12), 1554–1568.

Favela, J., & Peña-Mora, F. (2001). An experience in collaborative software engineering education. *Software, IEEE*, *18*(2), 47–53.

Ferris, G. R., Liden, R. C., Munyon, T. P., Summers, J. K., Basik, K. J., & Buckley, M. R. (2009). Relationships at work: Toward a multidimensional conceptualization of dyadic work relationships. *Journal of Management*, *35*, 1379–1403.

Fiore, S. M., Salas, E., & Cannon-Bowers, J. A. (2001). Group dynamics and shared mental model development. In M. London (Ed.), *How people evaluate others in organizations* (pp. 309–336). Mahwah, NJ: Lawrence Erlbaum.

Fishbach, A., & Ferguson, M. (1996). The goal construct in social psychology. In A. W. Kruglanski & T. E. Higgins (Eds.), *Social psychology: Handbook of basic principles* (pp. 490–515). New York, NY: Guilford Press.

Fiske, S. T. (1993). Social cognition and social perception. *Annual review of psychology*, *44*(1), 155–194.

Fitzgerald, B. (1997). The use of systems development methodologies in practice: A field study. *Information Systems Journal*, *7*(3), 201–212.

Fitzgerald, B. (1998). An empirical investigation into the adoption of systems development methodologies. *Information & Management*, *34*, 317–328.

Fitzgerald, B. (2000). Systems development methodologies: The problem of tenses. *Information Technology & People*, *13*, 174–185.

Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, *15*(2), 200–213.

Fitzgerald, B., Russo, N. L., & Stolterman, E. (2002). *Information systems development: Methods in action*. New York, NY: McGraw-Hill.

Fitzgerald, G., Philippides, A., & Probert, S. (1999). Information systems development, maintenance and enhancement: Findings from a UK study. *International Journal of Information Management*, *19*(4), 319–328.

Fransella, F., Bell, R., & Bannister., D. (2004). *A manual for repertory grid technique* (2nd ed.). Hoboken, NJ: John Wiley & Sons.

Frijda, N. H., Manstead, A. S. R., & Bem, S. (2000). The influence of emotions on beliefs. In *Emotions and beliefs: How feelings influence thoughts*. London, UK: Cambridge University Press.

Gartner. (2013). *Hunting and harvesting in a digital world: The 2013 CIO agenda*. Stamford, Connecticut. Retrieved from http://www.gartner.com/imagesrv/cio/pdf/cio_agenda_insights2013.pdf

Gartner Research. (2010). IT key metrics database. Retrieved from http://www.gartner.com/technology/consulting/key_metrics_data.jsp

Gilbert, S. (2004). Wearing two hats: Analyst-managers for small software projects. *IT Professional*, *6*, 34–39.

Gioia, D. A., & Chittipeddi, K. (1991). Sensemaking and sensegiving in strategic change initiation. *Strategic Management Journal*, *12*(6), 433–448.

Gittell, J. H. (2001). Supervisory span, relational coordination and flight departure performance: A reassessment of postbureaucracy theory. *Organization Science*, *12*(4), 468–483.

Gittell, J. H. (2011). New directions for relational coordination theory. In K. S. Cameron & G. Spreitzer (Eds.), *Oxford handbook of positive organizational scholarship* (pp. 74–94). Oxford, UK: Oxford University Press.

Glass, R. L. (2003). *Facts and fallacies of software engineering*. Boston, MA: Addison-Wesley Professional.

Gouldner, A. W. (1960). The norm of reciprocity: A preliminary statement. *American Sociological Review*, *25*(2), 161–178.

Graen, G. B., & Uhl-Bien, M. (1995). Development of leader-member exchange (LMX) theory of leadership over 25 years: Applying a multi-level multi-domain perspective. *Leadership Quarterly*, *6*, 219–247.

Green, S. G., Anderson, S. E., & Shivers, S. L. (1996). Demographic and organizational influences on leader–member exchange and related work attitudes. *Organizational Behavior and Human Decision Processes*, *66*(2), 203–214.

Groome, D. (2006). *An introduction to cognitive psychology: Processes and disorders* (2nd ed.). Hove, England: Psychology Press.

Guinan, P. J., Cooprider, J. G., & Faraj, S. (1998). Enabling software development team performance during requirements definition: A behavioral versus technical approach. *Information Systems Research*, *9*(2), 101–125.

Gurtner, A., Tschan, F., Semmer, N. K., & Nägele, C. (2007). Getting groups to develop good strategies: Effects of reflexivity interventions on team process, team performance, and shared mental models. *Organizational Behavior and Human Decision Processes*, *102*(2), 127–142.

Guzman, I. R., Stam, K. R., & Stanton, J. M. (2008). The occupational culture of IS/IT personnel within organizations. *The DATA BASE for Advances in Information Systems*, *39*(1), 33–50. doi:10.1145/1341971.1341976

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2009). *Multivariate data analysis* (7th ed.). New York, NY: Prentice Hall.

Halbesleben, J. R. (2011). Positive coworker exchanges. In E. Boos, M. Kolbe, P. M. Kappeler, & T. Ellwart (Eds.), *Coordination in human and primate groups* (pp. 107–130). New York, NY: Springer.

Hamilton, D. L. (2005). Social cognition: An introductory overview. In D. L. Hamilton (Ed.), *Social cognition: Key readings* (pp. 1–26). New York: Psychology Press.

Hardgrave, B. C., Davis, F. D., & Riemenschneider, C. K. (2003). Investigating determinants of software developers' intentions to follow methodologies. *Journal of Management Information Systems*, *20*(1), 123–152.

Harrison, D. A., Price, K. H., & Bell, M. P. (1998). Beyond relational demography: Time and the effects of surface- and deep-level diversity on work group cohesion. *Academy of Management Journal*, *41*(1), 96–107. doi:10.2307/256901

Haslam, S. A., Powell, C., & Turner, J. (2000). Social identity, self-categorization, and work motivation: Rethinking the contribution of the group to positive and sustainable organisational outcomes. *Applied Psychology*, *49*(3), 319–339.

Hassenzahl, M., & Wessler, R. (2000). Capturing design space from a user perspective: The repertory grid technique revisited. *International Journal of Human-Computer Interaction*, *12*(3-4), 441–459.

He, J., Butler, B., & King, W. (2007). Team cognition: Development and evolution in software project teams. *Journal of Management Information Systems*, *24*(2), 261–292. doi:10.2753/MIS0742-1222240210

Henderson, J. C., & Cooprider, J. G. (1990). Dimensions of I/S planning and design aids: A functional model of CASE technology. *Information Systems Research*, *1*(3), 227–254.

Highhouse, S. (2001). Judgment and decision-making research: relevance to industrial and organizational psychology. In N. Anderson, D. S. Ones, H. K. Sinangil, & C. Viswesvaran (Eds.), *Handbook of industrial, work and organizational psychology* (Vol. 2 - Organizational Psychology, pp. 314–331). London, UK: Sage Publications.

Highsmith, J. (2002). *Agile software development ecosystems*. Boston, MA: Addison-Wesley Professional.

Highsmith, J. (2009). *Agile project management: Creating innovative products*. Boston, MA: Addison-Wesley Professional.

Hodgkinson, G. P., Bown, N. J., Maule, A. J., Glaister, K. W., & Pearman, A. D. (1999). Breaking the frame: An analysis of strategic cognition and decision making under uncertainty. *Strategic Management Journal*, *20*(10), 977–985.

Hodgkinson, G. P., & Clarkson, G. P. (2005). What have we learned from almost 30 years of research on causal mapping? In V. Narayanan & D. J. Armstrong (Eds.), *Causal mapping for research in information technology* (pp. 46–79). Hershey, PA: Ideas group publishing.

Hodgson, D. (2002). Disciplining the professional: the case of project management. *Journal of Management Studies*, *39*(6), 803–821.

Hogg, M. A. (2001). A social identity theory of leadership. *Personality and Social Psychology Review*, *5*(3), 184–200.

Hogg, M. A., & Terry, D. J. (2000). Social identity and self-categorization processes in organizational contexts. *Academy of Management Review*, *25*(1), 121–140.

Hsu, J., Chang, J., Klein, G., & Jiang, J. (2011). Exploring the impact of team mental models on information utilization and project performance in system development. *International Journal of Project Management*, *29*(1), 1–12.

Huber, G. P., & Lewis, K. (2010). Cross-understanding: Implications for group cognition and performance. *Academy of Management Review*, *35*(1), 6–26. doi:10.5465/AMR.2010.45577787

Huisman, M., & Iivari, J. (2006). Deployment of systems development methodologies: Perceptual congruence between IS managers and systems developers. *Information & Management*, *43*(1), 29–49. doi:10.1016/j.im.2005.01.005

Humphrey, W. S., & Thomas, W. R. (2010). *Reflections on management: How to manage your software projects, your teams, your boss, and yourself*. Boston, MA: Addison-Wesley Professional.

Hunter, M. G., & Beck, J. E. (2000). Using repertory grids to conduct cross-cultural information systems research. *Information Systems Research*, *11*(1), 93–101.

Iivari, J., Hirschheim, R., & Klein, H. K. (1998). A paradigmatic analysis contrasting information systems development approaches and methodologies. *Information Systems Research*, *9*(2), 164–193.

Iivari, J., Hirschheim, R., & Klein, H. K. (2000). A dynamic framework for classifying information systems development methodologies and approaches. *Journal of Management Information Systems*, *17*(3), 179–218.

Iivari, J., Hirschheim, R., & Klein, H. K. (2004). Towards a distinctive body of knowledge for Information Systems experts: Coding ISD process knowledge in two IS journals. *Information Systems Journal*, *14*(4), 313–342. doi:10.1111/j.1365-2575.2004.00177.x

Iivari, J., & Huisman, M. (2007). The relationship between organisational culture and the deployment of systems development methodologies. *MIS Quarterly*, *31*(1), 35–58.

Iivari, J., & Iivari, N. (2011). The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, *53*(5), 509–520. doi:10.1016/j.infsof.2010.10.008

Irani, Z., Sharif, A., Love, P. E., & Kahraman, C. (2002). Applying concepts of fuzzy cognitive mapping to model: The IT/IS investment evaluation process. *International Journal of Production Economics*, *75*(1), 199–211.

Jankowicz, D. (2004). *The easy guide to repertory grids*. Hoboken, NJ: John Wiley & Sons.

Jayaratna, N. (1994). *Understanding and evaluating methodologies: NIMSAD, a systematic framework*. New York, NY: McGraw-Hill, Inc.

Jehn, K. A., Chadwick, C., & Thatcher, S. M. (1997). To agree or not to agree: The effects of value congruence, individual demographic dissimilarity, and conflict on workgroup outcomes. *International Journal of Conflict Management*, *8*(4), 287–305.

Jehn, K. A., Northcraft, G. B., & Neale, M. A. (1999). Why differences make a difference: A field study of diversity, conflict and performance in workgroups. *Administrative Science Quarterly*, *44*(4), 741–763. doi:10.2307/2667054

Jenkin, T. A., & Chan, Y. E. (2009). IS project alignment–a process perspective. *Journal of Information Technology*, *25*(1), 35–55.

Jiang, J. J., Klein, G., & Chen, H. G. (2001). The relative influence of IS project implementation policies and project leadership on eventual outcomes. *Project Management Journal*, *32*(3), 49–55.

Jiang, J. J., Sobol, M. G., & Klein, G. (2000). Performance ratings and importance of performance measures for IS staff: The different perceptions of IS users and IS staff. *Engineering Management, IEEE Transactions on*, *47*(4), 424–434.

Johri, A., & Nair, S. (2011). The role of design values in information system development for human benefit. *Information Technology & People*, *24*(3), 281–302.

Joshi, K. D., Sarker, S., & Sarker, S. (2007). Knowledge transfer within information systems development teams: Examining the role of knowledge source attributes. *Decision Support Systems*, *43*(2), 322–335.

Kankanhalli, A., Tan, B. C. Y., Wei, K. K., & Holmes, M. C. (2004). Cross-cultural differences and information systems developer values. *Decision Support Systems*, *38*, 183–195.

Karlsson, F., & Agerfalk, P. (2009). Exploring agile values in method configuration. *European Journal of Information Systems*, *18*(4), 300–316.

Kast, F. E., & Rosenzweig, J. E. (1972). General systems theory: Applications for organization and management. *Academy of Management Journal*, *15*(4), 447–465.

Keen, P. G. (1980). *MIS research: Reference disciplines and a cumulative tradition*. Center for Information Systems Research, Alfred P. Sloan School of Management.

Keil, F. C. (1992). *Concepts, kinds, and cognitive development*. Cambridge, MA: The MIT Press.

Keil, M., Mann, J., & Rai, A. (2000). Why software projects escalate: An empirical analysis and test of four theoretical models 1, 2. *MIS Quarterly*, *24*(4), 631–664.

Keil, M., Tan, B. C. Y., Wei, K. K., Saarinen, T., Tuunainen, V., & Wassenaar, A. (2000). A cross-cultural study on escalation of commitment behavior in software projects. *MIS Quarterly*, *24*(2), 299–325.

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

Kelly, G. (1955). *The psychology of personal constructs*. London, UK: Norton.

Kendall, K. E., & Kendall, J. E. (2004). *Systems Analysis and Design* (6th ed.). New Jersey, USA: Prentice Hall.

Kensing, F., & Munk-Madsen, A. (1993). PD: Structure in the toolbox. *Communications of the ACM*, *36*(4), 78–85.

Kim, T.-Y., Cable, D. M., & Kim, S.-P. (2005). Socialization tactics, employee proactivity, and person-organization fit. *Journal of Applied Psychology*, *90*(2), 232.

Kirsch, L. S. (1996). The management of complex tasks in organizations: Controlling the systems development process. *Organization Science*, *7*(1), 1–21.

Kirsch, L. S. (1997). Portfolios of control modes and IS project management. *Information Systems Research*, *8*(3), 215–239.

Klimoski, R., & Donahue, L. M. (2001). Person perception in organizations: An overview of the field. In M. London (Ed.), *How people evaluate others in organizations* (pp. 5–43). Mahwah, NJ: Lawrence Erlbaum.

Klimoski, R., & Mohammed, S. (1994). Team mental model: Construct or metaphor? *Journal of Management*, *20*(2), 403–437.

Klohnen, E. C., & Luo, S. (2003). Interpersonal attraction and personality: What is attractive-self similarity, ideal similarity, complementarity or attachment security? *Journal of Personality and Social Psychology*, *85*(4), 709–722.

Ko, D. G., Kirsch, L. S., & King, W. R. (2005). Antecedents of knowledge transfer from consultants to clients in enterprise system implementations. *MIS Quarterly*, *29*(1), 59–85.

Koh, C., Ang, S., & Straub, D. W. (2004). IT outsourcing success: A psychological contract perspective. *Information Systems Research*, *15*(4), 356–373.

Krauss, R., & Fussell, S. (1990). Mutual knowledge and communicative effectiveness. In J. Galagher, R. Kraut, & C. Egdio (Eds.), *Intellectual teamwork: Social and technological foundations of cooperative work*. Hillsdale, NJ: Lawrence Erlbaum.

Krishna, V., & Basu, A. (2011). Scrum+: Is it "ScrumBut" or "ScrumAnd." In *India Conference (INDICON), 2011 Annual IEEE* (pp. 1–4).

Kristof, A. L. (1996). Person-organization fit: An integrative review of its conceptualizations, measurement, and implications. *Personnel Psychology*, *49*, 1–49.

Kristof-Brown, A. L., Zimmerman, R. D., & Johnson, E. C. (2005). Consequences of individuals'fit at work: A meta-analysis of person-job, person-organization, person-group, and person-supervisor fit. *Personnel Psychology*, *58*, 281–342.
Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*(1), 1–27.
Kudikyala, U. K., & Vaughn, R. B. (2005). Software requirement understanding using Pathfinder networks: Discovering and evaluating mental models. *Journal of Systems and Software*, *74*(1), 101–108. doi:10.1016/j.jss.2003.09.028
Kumar, K., & Bjorn-Andersen, N. (1990). A cross-cultural comparison of IS designer values. *Communications of the ACM*, *33*, 528–538.
Kunda, Z. (1999). *Social cognition: Making sense of people*. Cambridge, MA: The MIT Press.
Langan-Fox, J., Code, S., & Langfield-Smith, K. (2000). Team mental models: Techniques, methods, and analytic approaches. *Human Factors*, *42*(2), 242–271. doi:10.1518/001872000779656534
Lau, D. C., & Murnighan, J. K. (1998). Demographic diversity and faultlines: The compositional dynamics of organizational groups. *Academy of Management Review*, *23*(2), 325–340. doi:10.5465/AMR.1998.533229
Lau, W. W. F., & Yuen, A. H. K. (2010). Promoting conceptual change of learning sorting algorithm through the diagnosis of mental models: The effects of gender and learning styles. *Computers & Education*, *54*(1), 275–288. doi:10.1016/j.compedu.2009.08.014
Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, *34*(1), 87–114.
Lee, J., Graen, G. B., & Graen, J. (2005). *Communication as antecedents and consequences of LMX development globally: A new strong inference approach*. Greenwich, CT: Information Age Pub.
Lee, J., & Truex, D. P. (2000). Exploring the impact of formal training in ISD methods on the cognitive structure of novice information systems developers. *Information Systems Journal*, *10*, 347–367.
Lee, T. W. (1998). *Using qualitative methods in organizational research*. London, UK: Sage Publications Ltd.
Levesque, L., Wilson, J., & Wholey, D. (2001). Cognitive divergence and shared mental models in software development project teams. *Journal of Organizational Behavior*, *22*, 135–144.
Levin, D. Z., & Cross, R. (2004). The strength of weak ties you can trust: The mediating role of trust in effective knowledge transfer. *Management Science*, *50*(11), 1477–1490.
Levina, N., & Vaast, E. (2005). The emergence of boundary spanning competence in practice: Implications for implementation and use of information systems. *MIS Quarterly*, *29*(2), 335–363.
Lewick, R. J., & Bunker, B. B. (1996). Developing and maintaining trust in work relationships. In *Trust in Organizations: Frontiers of Theory and Reach* (pp. 114–39). Thousand Oaks, CA: SAGE Publications.
Lewis, B. R., Templeton, G. F., & Byrd, T. A. (2005). A methodology for construct development in MIS research. *European Journal of Information Systems*, *14*, 388–400.
Liang, T. P., Jiang, J., Klein, G. S., & Liu, J. Y. C. (2010). Software quality as influenced by informational diversity, task conflict, and learning in project teams. *Engineering Management, IEEE Transactions on*, *57*(3), 477–487.

Liang, T. P., Liu, C.-C., Lin, T.-M., & Lin, B. (2007). Effect of team diversity on software project performance. *Industrial Management & Data Systems*, *107*(5), 636 – 653.

Lim, B. C., & Klein, K. J. (2006). Team mental models and team performance: A field study of the effects of team mental model similarity and accuracy. *Journal of Organizational Behavior*, *27*(4), 403–418. doi:10.1002/job.387

Lofquist, L. H., & Dawis, R. V. (1978). Values as second-order needs in the theory of work adjustment. *Journal of Vocational Behavior*, *12*, 12–19.

Lovelace, K., Shapiro, D. L., & Weingart, L. R. (2001). Maximizing cross-functional new product teams' innovativeness and constraint adherence: A conflict communications perspective. *Academy of management journal*, *44*(4), 779–793.

Lührmann, T., & Eberl, P. (2007). Leadership and identity construction: Reframing the leader-follower interaction from an identity theory perspective. *Leadership*, *3*(1), 115–127. doi:10.1177/1742715007073070

MacCorquodale, K., & Meehl, P. E. (1948). On a distinction between hypothetical constructs and intervening variables. *Psychological Review*, *55*(2), 95–107.

Maddison, R., Baker, G., Bhabuta, L., Fitzgerald, G., Hindle, K., Song, J., … Wood, J. (1984). Feature analysis of five information system methodologies. In T. Bemelmans (Ed.), *Beyond Productivity: Information Systems Development for Organisational Effectiveness*. Amsterdam, Netherlands: North Holland Publishing.

Mahaney, R. C., & Lederer, A. L. (2003). Information systems project management: An agency theory interpretation. *Journal of Systems and Software*, *68*, 1–9.

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, *26*(1), 87–119.

March, J. G., & Simon, H. A. (1958). *Organizations.* Oxford, England: Wiley.

Markman, A. B., & Ross, B. H. (2003). Category use and category learning. *Psychological Bulletin*, *129*, 592–613.

Markoczy, L., & Goldberg, J. (1995). A method for eliciting and comparing causal maps. *Journal of Management*, *21*(2), 305–333.

Marks, M. A., Sabella, M. J., Burke, C. S., & Zaccaro, S. J. (2002). The impact of cross-training on team effectiveness. *Journal of Applied Psychology*, *87*(1), 3–13. doi:10.1037/0021-9010.87.1.3

Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, *20*(3), 377–399.

Mathieu, J. E., Heffner, T. S., Goodwin, G. F., Salas, E., & Cannon-Bowers, J. A. (2000). The influence of shared mental models on team process and performance. *Journal of Applied Psychology*, *85*, 273–283.

Maxwell, K. D., & Forselius, P. (2000). Benchmarking software development productivity. *Software, IEEE*, *17*, 80–88.

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of management review*, *20*(3), 709–734.

Mayo, E. (1933). *The human problems of an industrial civilization*. New York, NY: Macmillan.

McAllister, D. J. (1995). Affect-and cognition-based trust as foundations for interpersonal cooperation in organizations. *Academy of management journal*, *38*(1), 24–59.

McKeen, J. D., Guimaraes, T., & Wetherbe, J. C. (1994). The relationship between user participation and user satisfaction: An investigation of four contingency factors. *MIS Quarterly*, *18*(4), 427–451.

Medin, D. L. (1989). Concepts and conceptual structure. *American Psychologist; American Psychologist*, *44*, 1469 – 1481.

Medin, D. L., Lynch, E. B., & Solomon, K. O. (2000). Are there kinds of concepts? *Annual Review of Psychology*, *51*(1), 121–147.

Michalski, R. S. (1993). Inferential theory of learning as a conceptual basis for multistrategy learning. *Machine Learning*, *11*, 111–151.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2013). *Qualitative data analysis: A methods sourcebook* (3rd ed.). London, UK: SAGE Publications.

Miller, D., & Toulouse, J. M. (1986). Chief executive personality and corporate strategy and structure in small firms. *Management Science*, *32*, 1389–1409.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, *63*, 81–97.

Mohammed, S., Ferzandi, L., & Hamilton, K. (2010). Metaphor no more: A 15-Year review of the team mental model construct. *Journal of Management*, *36*(4), 876–910. doi:10.1177/0149206309356804

Mohammed, S., Klimoski, R., & Rentsch, J. R. (2000). The measurement of team mental models: We have no shared schema. *Organizational Research Methods*, *3*(2), 123–165.

Montazemi, A. R., & Conrath, D. W. (1986). The use of cognitive mapping for information requirements analysis. *MIS Quarterly*, *10*(1), 45–56.

Moser, P. K. (1995). Epistemology. In R. Audi (Ed.), *The Cambridge dictionary of philosophy* (pp. 233–238). Cambridge, UK: Cambridge University Press.

Moynihan, T. (1996). An inventory of personal constructs for information systems project risk researchers. *Journal of Information Technology*, *11*(4), 359–371.

Moynihan, T. (1997). How experienced project managers assess risk. *Software, IEEE*, *14*(3), 35–41.

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

Mumford, E. (1995). *Effective systems design and requirements analysis: the ETHICS approach*. Basingstoke, UK: Macmillan.

Murch, R. (2001). *Project management: Best practices for IT professionals*. Upper Saddle River, NJ: Prentice Hall.

Nahapiet, J., & Ghoshal, S. (1998). Social capital, intellectual capital, and the organizational advantage. *Academy of Management Review*, *23*(2), 242–266.

Napier, N. P., Keil, M., & Tan, F. B. (2009). IT project managers' construction of successful project management practice: A repertory grid investigation. *Information Systems Journal*, *19*(3), 255–282. doi:doi:10.1111/j.1365-2575.2007.00264.x

Nelson, H. J., Armstrong, D. J., & Ghods, M. (2002). Old dogs and new tricks. *Communications of the ACM*, *45*(10), 132–137.

Nelson, K. M., Nadkarni, S., Narayanan, V., & Ghods, M. (2000). Understanding software operations support expertise: a revealed causal mapping approach. *MIS Quarterly*, *24*(3), 475–507.

Nelson, R. R. (1991). Educational needs as perceived by IS and end-user personnel: A survey of knowledge and skill requirements. *MIS Quarterly*, *15*(4), 503–525.

Nerur, S., Mahapatra, R. K., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72–78.
Ngwenyama, O., & Nielsen, P. A. (2003). Competing values in software process improvement: an assumption analysis of CMM from an organizational culture perspective. *IEEE Transactions on Engineering Management*, *50*, 100–112.
Nidumolu, S. R., & Subramani, M. R. (2003). The matrix of control: Combining process and structure approaches to managing software development. *Journal of Management Information Systems*, *20*(3), 159–196.
Niu, N., & Easterbrook, S. (2006). Discovering aspects in requirements with repertory grid. In *Proceedings of the 2006 international workshop on Early aspects at ICSE* (pp. 35–42). ACM.
Nystrom, P. C., & Starbuck, W. H. (1984). Managing beliefs in organizations. *Journal of Applied Behavioral Science*, *20*(3), 277–287. doi:10.1177/002188638402000308
Olson, D. L. (2001). *Introduction to information systems project management*. New York, NY: Irwin/McGraw-Hill.
Orasanu, J. (1994). Shared problem models and flight crew performance. *Aviation psychology in practice*, 255–285.
Ouchi, W. G., & Maguire, M. A. (1975). Organizational control: Two functions. *Administrative Science Quarterly*, *20*, 559–569.
Ozer, M. (2008). Personal and task-related moderators of leader-member exchange among software developers. *Journal of Applied Psychology*, *93*(5), 1174–1182.
Pankratz, O., & Loebbecke, C. (2011). Project managers' perception of is project success factors: A repertory grid investigation. *ECIS 2011 Proceedings*, 170–182.
Park, C. L. (2010). Making sense of the meaning literature: an integrative review of meaning making and its effects on adjustment to stressful life events. *Psychological bulletin*, *136*(2), 257.
Passos, C., Braun, A. P., Cruzes, D. S., & Mendonça, M. (2011). Analyzing the impact of beliefs in software project practices. In *International Symposium on Empirical Software Engineering and Measurement* (pp. 444–452).
Patton, M. Q. (2002). *Qualitative research and evaluation methods* (3rd ed.). London: Sage Publications Ltd.
Pikkarainen, M., & Passoja, U. (2005). An approach for assessing suitability of agile solutions: a case study. In *Extreme Programming and Agile Processes in Software Engineering* (pp. 171–179). Springer.
Podsakoff, P. M., MacKenzie, S. B., Paine, J. B., & Bachrach, D. G. (2000). Organizational citizenship behaviors: A critical review of the theoretical and empirical literature and suggestions for future research. *Journal of Management*, *26*, 513.
Politis, J. D. (2003). The connection between trust and knowledge management: What are its implications for team performance. *Journal of Knowledge Management*, *7*(5), 55–66.
Poppendieck, M., & Poppendieck, T. (2003). *Lean software development: An agile toolkit*. Boston, MA: Addison-Wesley Professional.
Preston, D. S., & Karahanna, E. (2009). Antecedents of IS strategic alignment: A nomological network. *Information Systems Research*, *20*(2), 159–179.
Project Management Institute. (2008). *A guide to the project management body of knowledge: PMBOK® guide* (4th ed.). Newtown, PA: Project Management Institute.

Pruitt, D. G., & Robin, J. Z. (1986). *Social conflict: Escalation, stalemate, and settlement*. New York, NY: Random House.
Quinn, R. E., & Rohrbaugh, J. (1981). A competing values approach to organizational effectiveness. *Public Productivity Review*, 122–140.
Quinn, R. E., & Rohrbaugh, J. (1983). A spatial model of effectiveness criteria: Towards a competing values approach to organizational analysis. *Management Science*, *29*(3), 363–377.
Ragins, B. R., & Cotton, J. L. (1999). Mentor functions and outcomes: A comparison of men and women in formal and informal mentoring relationships. *Journal of Applied Psychology*, *84*(4), 529–550. doi:10.1037/0021-9010.84.4.529
Ramachandran, S., & Rao, S. V. (2006). An effort towards identifying occupational culture among information systems professionals (pp. 198–204). Presented at the Special Interest Group on Computer Personnel Research Annual Conference, ACM New York, NY, USA.
Ramesh, B., Mohan, K., & Cao, L. (2012). Ambidexterity in Agile distributed development: An empirical investigation. *Information Systems Research*, *23*(2), 323–339.
Randolph-Seng, B., & Norris, J. I. (2011). Cross-understanding in groups: How to "cross over" without "dying." *Academy of Management Review*, *36*(2), 420–422.
Ravichandran, T., & Lertwongsatien, C. (2005). Effect of information systems resources and capabilities on firm performance: A resource-based perspective. *Journal of Management Information Systems*, *21*(4), 237–276.
Reed, S. K. (2007). *Cognition: Theory and applications*. Belmont, CA: Wadsworth Pub Co.
Reger, R. K. (1990). The repertory grid technique for eliciting the content and structure of cognitive constructive systems. In S. A (Ed.), *Mapping Strategic Thought* (pp. 301–309). Chicester, UK: John Wiley & Sons.
Reich, B. H., & Benbasat, I. (2000). Factors that influence the social dimension of alignment between business and information technology objectives. *MIS Quarterly*, *24*(1), 81–114.
Rentsch, J. R., & Zelno, J. A. (2005). The role of cognition in managing conflict to maximize team effectiveness. In M. A. West, D. Tjosvold, & K. G. Smith (Eds.), *The essentials of teamworking: International perspectives* (pp. 71–90). West Sussex, UK: John Wiley & Sons Ltd.
Rico, R., Sánchez-Manzanares, M., Gil, F., & Gibson, C. (2008). Team implicit coordination processes: A team knowledge-based approach. *Academy of Management Review*, *33*, 163–184.
Riemenschneider, C. K., Hardgrave, B. C., & Davis, F. D. (2002). Explaining software developer acceptance of methodologies: A comparison of five theoretical models. *Software Engineering, IEEE Transactions on*, *28*(12), 1135–1145.
Roberson, M. T., & Sundstrom, E. (1990). Questionnaire design, return rates, and response favorableness in an employee attitude questionnaire. *Journal of Applied Psychology*, *75*(3), 354.
Roberts, E. B., & Fusfeld, A. I. (1997). *Informal leadership role in the innovation process*. New York, NY: Oxford University Press.
Rokeach, M. (1960). *The open and closed mind*. Oxford, England: Basic Books.
Rouse, W. B., Cannon-Bowers, J. A., & Salas, E. (1992). The role of mental models in team performance in complex systems. *Systems, Man and Cybernetics, IEEE Transactions on*, *22*(6), 1296–1308.

Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: Prospects and limits in the search for mental models. *Psychological Bulletin*, *100*(3), 349–363.
Schneider, B. (1987). The people make the place. *Personnel psychology*, *40*, 437–453.
Schroder, H. M., Driver, M. J., & Streufert, S. (1967). *Human information processing: Individuals and groups functioning in complex social situations*. New York, NY: Holt, Rinehart & Winston .
Schuelke, M. J., Day, E. A., McEntire, L. E., Boatman, P. R., Boatman, J. E., Kowollik, V., & Wang, X. (2009). Relating indices of knowledge structure coherence and accuracy to skill-based performance: Is there utility in using a combination of indices? *Journal of Applied Psychology*, *94*, 1076–1085.
Schwaber, K., & Beedle, M. (2001). *Agile software development with Scrum*. Upper Saddle River,NJ: Prentice Hall.
Schwartz, S. H. (1992). Universals in the content and structure of values: Theory and empirical tests in 20 countries. In M. Zanna (Ed.), *Advances in experimental social psychology* (Vol. 25, pp. 1–65). New York, NY: Academic Press.
SEI, C. (1994). *The capability maturity model: Guidelines for improving the software process*. Reading, MA: Addison-Wesley.
Serfaty, D., Entin, E. E., & Johnston, J. H. (1998). Team coordination training. In J. A. Cannon-Bowers & E. Salas (Eds.), *Making decisions under stress: Implications for individual and team training* (pp. 221–245). Washington, DC: American Psychological Association.
Serva, M. A., Fuller, M. A., & Mayer, R. C. (2005). The reciprocal nature of trust: A longitudinal study of interacting teams. *Journal of organizational behavior*, *26*(6), 625–648.
Settoon, R. P., & Mossholder, K. W. (2002). Relationship quality and relationship context as antecedents of person-and task-focused interpersonal citizenship behavior. *Journal of Applied Psychology*, *87*(2), 255.
Shaft, T. M., Albert, L. J., & Jasperson, J. S. (2008). Managing change in an information systems development organization: Understanding developer transitions from a structured to an object-oriented development environment1. *Information Systems Journal*, *18*(3), 275–297. doi:10.1111/j.1365-2575.2007.00270.x
Shaft, T. M., & Vessey, I. (2006). The role of cognitive fit in the relationship between software comprehension and modification. *MIS Quarterly*, *30*(1), 29–55.
Sheetz, S. D., & Tegarden, D. P. (2001). Illustrating the cognitive consequences of object-oriented systems development. *Journal of Systems and Software*, *59*(2), 163–179.
Shore, B., & Venkatachalam, A. R. (1995). The role of national culture in systems analysis and design. *Journal of Global Information Management*, *3*, 5–14.
Siakas, K. V., & Siakas, E. (2007). The agile professional culture: A source of agile quality. *Software Process: Improvement and Practice*, *12*, 597–610.
Siau, K., & Tan, X. (2005a). Improving the quality of conceptual modeling using cognitive mapping techniques. *Data & Knowledge Engineering*, *55*(3), 343–365.
Siau, K., & Tan, X. (2005b). Technical communication in information systems development: The use of cognitive mapping. *IEEE Transactions on Professional Communication*, *48*, 269–284.
Siau, K., & Tan, X. (2006). Using cognitive mapping techniques to supplement UML and UP in information requirements determination. *Journal of Computer Information Systems*, *47*, 59–66.

Siau, K., & Tan, X. (2006). Cognitive mapping techniques for user-database interaction. *Professional Communication, IEEE Transactions on*, *49*(2), 96–108.
Siau, K., & Tan, X. (2008). Use of cognitive mapping techniques in information systems development. *Journal of Computer Information Systems*, *48*(4), 49–57.
Simmel, G. (1964). *The sociology of Georg Simmel. Translated, edited, and with an introduction by Kurt H. Wolff*. London, UK: Collier-Macmillan.
Smith, E. E. (1990). Categorization. In D. N. Osherson & E. E. Smith (Eds.), *Thinking: An invitation to cognitive science*. Cambridge, MA: MIT Press.
Smith-Jentsch, K. A. (2009). Measuring team-related cognition: The devil is in the details. In E. Salas, G. F. Goodwin, & C. S. Burke (Eds.), *Team effectiveness in complex organizations* (pp. 491–508). New York, NY: Routledge, Taylor & Francis Group.
Smith-Jentsch, K. A., Mathieu, J. E., & Kraiger, K. (2005). Investigating linear and interactive effects of shared mental models on safety and efficiency in a field setting. *Journal of Applied Psychology*, *90*(3), 525–535.
Sobek, D. K., Ward, A. C., & Liker, J. K. (1999). Toyota's principles of set-based concurrent engineering. *Sloan Management Review*, *40*(2), 67–84.
Sommerville, I. (2004). *Software engineering*. Boston, MA: Addison Wesley.
Sparrow, J. (1998). *Knowledge in organizations*. London, UK: Sage Publications.
Spence, I., & Ogilvie, J. C. (1973). A table of expected stress values for random rankings in nonmetric multidimensional scaling. *Multivariate Behavioral Research*, *8*(4), 511–517.
Standish Group. (2009). Standish Group chaos summary 2009. Retrieved from http://www1.standishgroup.com/newsroom/chaos_2009.php
Stewart, V., & Stewart, A. (1981). *Business applications of repertory grid*. London, UK: McGraw-Hill.
Stout, R. J., Cannon-Bowers, J. A., Salas, E., & Milanovich, D. M. (1999). Planning, shared mental models, and coordinated performance: An empirical link is established. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *41*(1), 61–71.
Suddaby, R. (2010). Editor's comments: Construct clarity in theories of management and organization. *Academy of Management Review*, *35*(3), 346–357. doi:10.5465/AMR.2010.51141319
Super, D. E. (1970). *Work values inventoy*. Boston, MA: Houghron-Mifflin.
Tan, F. B., & Hunter, M. G. (2002). The repertory grid technique: A method for the study of cognition in information systems. *MIS Quarterly*, *26*(1), 39–57.
Tan, F. B., Tung, L.-L., & Xu, Y. (2009). A study of web-designers' criteria for effective business-to-consumer (B2C) websites using the repertory grid technique'. *Journal of Electronic Commerce Research*, *10*(3), 155–177.
Taylor, F. W. (1911). *The principles of scientific management*. New York, NY,: Harper & Brothers.
Thatcher, S. M. B., & Greer, L. L. (2008). Does it really matter if you recognize who I am? The implications of identity comprehension for individuals in work teams†. *Journal of Management*, *34*, 5–24.
Tiwana, A. (2004). An empirical study of the effect of knowledge integration on software development performance. *Information and Software Technology*, *46*(13), 899–906.
Tiwana, A., & Keil, M. (2009). Control in internal and outsourced software projects. *Journal of Management Information Systems*, *26*(3), 9–44.

Tolfo, C., & Wazlawick, R. S. (2008). The influence of organizational culture on the adoption of extreme programming. *Journal of Systems and Software*, *81*, 1955–1967.
Tossell, C. C., Smith, B. A., & Schvaneveldt, R. W. (2009). The influence of rating method on knowledge structures. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 53, pp. 1893–1897). SAGE Publications.
Tulving, E. (1985). How many memory systems are there? *American psychologist*, *40*(4), 385.
Turner, J. C. (1987). *Rediscovering the social group: A self-categorization theory*. Oxford, UK: Basil Blackwell.
Turner, J. C., Brown, R. J., & Tajfel, H. (1979). Social comparison and group interest in ingroup favouritism. *European Journal of Social Psychology*, *9*(2), 187–204. doi:10.1002/ejsp.2420090207
Uhl-Bien, M. (2006). Relational leadership theory: Exploring the social processes of leadership and organizing. *The Leadership Quarterly*, *17*(6), 654–676.
Uhl-Bien, M., & Graen, G. B. (1993). *Leadership-making in self-managing professional work teams: An empirical investigation*. West Orange, NJ: Leadership Library of America.
Van de Ven, A. H., Delbecq, A. L., & Koenig Jr, R. (1976). Determinants of coordination modes within organizations. *American sociological review*, 322–338.
Van Der Vegt, G., Emans, B., & Van De Vliert, E. (2000). Team members' affective responses to patterns of intragroup interdependence and job complexity. *Journal of Management*, *26*(4), 633–655.
Van Der Vegt, G. S., Van De Vliert, E., & Oosterhof, A. (2003). Informational dissimilarity and organizational citizenship behavior: The role of intrateam interdependence and team identification. *The Academy of Management Journal*, *46*(6), 715–727.
Van Knippenberg, D., De Dreu, C. K. W., & Homan, A. C. (2004). Work group diversity and group performance: An integrative model and research agenda. *Journal of Applied Psychology*, *89*(6), 1008–1022. doi:10.1037/0021-9010.89.6.1008
Venkatraman, N. (1989). The concept of fit in strategy research: Toward verbal and statistical correspondence. *Academy of Management Review*, *14*(3), 423–444.
Vlaar, P. W. L., van Fenema, P. C., & Tiwari, V. (2008). Cocreating understanding and value in distributed work: How members of onsite and offshore vendor teams give, make, demand, and break sense. *MIS Quarterly*, *32*(2), 227–255.
Voke. (2012). *Market snapshot report: Agile realities*. Minden, NV.
Von Bertalanffy, L. (1968). *General system theory*. New York, NY: George Braziller.
Wagner, R. V. (1975). Complementary needs, role expectations, interpersonal attraction, and the stability of working relationships. *Journal of Personality and Social Psychology*, *32*(1), 116–124. doi:10.1037/h0076843
Waller, M. J., Gupta, N., & Giambatista, R. C. (2004). Effects of adaptive behaviors and shared mental models on control crew performance. *Management Science*, *50*(11), 1534–1544.
Walsh, J. P. (1995). Managerial and organizational cognition: Notes from a trip down memory lane. *Organization Science*, *6*, 280–321.
Wason, P. C. (1960). On the failure to eliminate hypotheses in a conceptual task. *Quarterly journal of experimental psychology*, *12*, 129–140.
Webber, S. S., & Donahue, L. M. (2001). Impact of highly and less job-related diversity on work group cohesion and performance: A meta-analysis. *Journal of Management*, *27*(2), 141–162. doi:10.1177/014920630102700202

Wegner, D. M. (1995). A computer network model of human transactive memory. *Social Cognition*, *13*(3), 319–339.
Weick, K. E. (1988). Enacted Sensemaking in Crisis Situations. *Journal of Management Studies*, *25*(4), 305–317.
Whitehead, J. (2007). Collaboration in software engineering: A roadmap. In *Future of Software Engineering, 2007. FOSE'07* (pp. 214–225). IEEE.
Whitten, J. l., & Bentley, l. D. (2006). *Introduction to systems analysis and design*. New York, NY: McGraw-Hill.
Williams, L. (2010). Agile software development methodologies and practices. *Advances in Computers*, *80*, 1–44.
Winch, R. F., Ktsanes, T., & Ktsanes, V. (1954). The theory of complementary needs in mate-selection: An analytic and descriptive study. *American Sociological Review*, *19*(3), 241–249.
Wright, R. P. (2008). Eliciting cognitions of strategizing using advanced repertory grids in a world constructed and reconstructed. *Organizational Research Methods*, *11*(4), 753–769. doi:10.1177/1094428107303353
Xia, W., & Lee, G. (2003). Complexity of information systems development projects: Conceptualization and measurement development. *Journal of Management Information Systems*, *22*(1), 45–83.
Yang, H. D., Kang, H. R., & Mason, R. M. (2008). An exploratory study on meta skills in software development teams: Antecedent cooperation skills and personality for shared mental models. *European Journal of Information Systems*, *17*, 47–61. doi:10.1057/palgrave.ejis.3000730
Yin, R. K. (2009). *Case study research: Design and methods* (4th ed.). London, UK: Sage Publications Ltd.
Yu, Y., Hao, J.-X., Dong, X.-Y., & Khalifa, M. (2013). A multilevel model for effects of social capital and knowledge sharing in knowledge-intensive work teams. *International Journal of Information Management*, *33*(5), 780–790.
Yuan, M., Zhang, X., Chen, Z., Vogel, D. R., & Chu, X. (2009). Antecedents of coordination effectiveness of software developer dyads from interacting teams: an empirical investigation. *Engineering Management, IEEE Transactions on*, *56*(3), 494–507.
Zmud, R. W., Anthony, W. P., & Stair Jr, R. M. (1993). The use of mental imagery to facilitate information identification in requirements analysis. *Journal of Management Information Systems*, *9*(4), 175–191.

Appendix A

ISD Concepts – Content Analysis

ISD Concepts	Subordinate Concepts or Work Practices	ISD Methodology	ISD Approach
User involvement and participation	▪ User involvement in the test development and validation	eXtreme Programming (XP)	Agile
	▪ On-site dedicated customer (close, daily cooperation between business people and developers)	XP	Agile
	▪ Emphasizing participative development among system owners, users, designers, and builder	Evolutionary, Web development	
	▪ Prototyping	Evolutionary, Spiral	
	▪ The setting up of a steering committee and a design group	ETHICS (Effective Technical and Human Implementation of Computer-based Systems)	Socio-technical
	▪ The RAD approach recognizes that user involvement is necessary for intellectual reasons – to reduce costly requirements, for example – and for political reasons users may reject systems outright if they have not been sufficiently involved in development. However, the RAD operationalization of the user involvement concept is very rich one. At the heart of the RAD approach are Joint application		Rapid application development (RAD)

	development (JAD) and joint requirements planning (JRP). ▪ Joint requirement planning		
	▪ End-of-iteration customer focus group	RADical Software Development	Agile
	▪ Spend enough time in conversation to understand business processes and the customer's real problems ▪ Create effective working relationships between development staffs and their customers ▪ Client-driven iterative development		
	▪ Ensuring user/customer/marketing input at all stages of the project	Software Best Practice Survey (SBPQ)	
	▪ End-user computing		
	▪ Users are responsible for the design		
	▪ Determining the highest-priority set of requirements to be included in the next iteration is done collaboratively by the customers and developers		Agile
Close collaboration and communication between project members during development	▪ Pair programming	XP	Agile
	▪ The development team is located in a collaborative workspace - a space which supports and facilitates communication (synonyms: common project room, sit together) ▪ Get the programmers together with the person who has the problem ▪ Talk together at a whiteboard, make a decision, and then go and code it.	Scrum, XP	Agile
	▪ Scrum meetings (daily stand-up meetings)	Scrum	Agile
	▪ The walls of the development workspace serve as a communication		Agile

	means, constituting an informative and collaborative workspace. The information posted on the walls includes, among other relevant items, the status of the personal tasks that belong to the current iteration and the measures taken. Thus, all the project stakeholders can be updated at a glance at any time about the project's progress		
	▪ Face-to-face conversation is the best form of communication rather than "talking" through documents ▪ At the end of each iteration, the development team demonstrates the acceptance test for the functionality completed during the iteration to the development team, the customer, and other interest parties		Agile
	▪ Develop perfect communication, coordination, and collaboration policy across people and processes (Cobb, 2011).		Lean
	▪ Use the most effective communications and coordination practices and effective tools ▪ tTcit knowledge exchange		Lean
	▪ Metaphor: guide all development with a simple shared story of how the whole system works.	XP	Agile
	▪ Small team size ensures that the potential for communication distortion and conflict is kept to a minimum		Rapid application development (RAD)
	▪ All relevant parties in JAD and JRP workshops are co-located thus leading to synchronization in the communication process.		RAD
Iterative and incremental development	▪ Smaller releases of the system: Each release should be as small as possible while still containing a coherent set of business requirements. Generally short time-boxes of one or two months are	XP	Agile, RAD

	preferable to six months or a year.		
	▪ Incremental test development from scenarios	XP	Agile
	▪ Prototyping		Agile, RAD
	▪ Allows or accommodates future change by building in some flexibility or redundancy, or by parameterizing as much as possible	Evolutionary	
	▪ Not all a system's requirements can necessarily be identified and specified in advance. Thus, systems emerge though iterative prototyping, with iteration seen as useful and necessary, not as re-wok delaying development.		RAD
	▪ The requirements are so difficult to define for users and that often they cannot articulate or define in any detail their requirements		Agile
	▪ Structuring the implementation based upon current desire, reality, and knowledge rather than what was defined in a contract before the project began	XP	Agile
	▪ Invent in the design of the system every day in light of the experience of the past and the current needs		Agile
	▪ Planning does not need to be considered making a commitment. To prepare for inevitable change, defer critical design decisions until the last responsible moments.		Lean development
	▪ Frequent deadlines reduce the variance of an ISD process	Evolutionary	
	▪ Welcome changing requirements, even late in development		Agile
	▪ Iterative enhancement	Evolutionary	
	▪ Plan often when you can't plan accurately (team plans must change on a regular basis)	Humphrey and Thomas (2010)	
	▪ Risk-driven, evolving requirements specifications		Agile, CMMI
	▪ Rapid iteration cycles to determine needed changes in the desired		Agile

	capability and to fix them in the next iteration		
	▪ Develop software iteratively	RUP	
	▪ Periodic reprioritization of the backlog keeps the product on track over the long term.	Scrum	Agile
	▪ Post iteration reflection workshops keep the team and process on track over the long term	Scrum, Crystal Clear	Agile
Continuous integration	▪ The use of automated test	XP	Agile
	▪ Programmers load the latest release, load their changes, and run the tests until they pass 100 percent. This ensures that the responsibility for fixing errors is always obvious, as code is not integrated unless it is working 100 percent.	XP	Agile
	▪ Integrate and build the system every time a task is completed – this may be many times per day.	XP	Agile
	▪ Design for change is often wasted effort/design does not have to be perfect/design does not have to be comprehensive	XP, Evolutionary,	RAD
	▪ Automate the build	XP, Feature-driven Development (FDD)	Agile
	▪ Maintain a code repository	XP	Agile
	▪ Automate deployment	XP	Agile
	▪ Mechanism for performance of regression testing during and after initial implementation	SBPQ	
	▪ Ten-minute build	XP	Agile
	▪ Teams use a nightly build system with which the product is built and all automated tests are run overnight.		
	▪ Maintain a single source repository		
	▪ Developers commit the code frequently		

Continuous attention to technical excellence	▪ Every commit should build the mainline on an integration machine. If mainline build fails, the problem should be fixed right away rather than waiting for the nightly build ▪ Constant refactoring to improve code quality ▪ Programmers restructure the system, without removing functionality, to improve non-functional aspects (e.g., duplication of code, simplicity, flexibility). ▪ Code is owned by a particular developer, often based upon a developer's area of expertise. Due to more opportunities for specialization and pride in the code, the technical excellence is more likely to attain. ▪ Periodic technical reviews (both informal and regular scheduled)	XP	Agile
	▪ Incremental rearchitecture	Crystal Clear	
	▪ Walking Skeleton	Crystal Clear	
Reuse of code and components	▪ Structuring the system architecture into components	Component-based development	RAD
	▪ Searching for shortcuts, reuse code, clone existing code and modify it, or utilize commercial package ▪ Reusing the code within the organization or buy from outside		
Standards for design	▪ Using graphical models to present static and dynamic views of the software		Object-oriented methods
	▪ Usage of design notation like Structured Analysis and Design Technique (SADT) ▪ Application of common coding standards or a coding style guideline	SBPQ	

Careful and comprehensive documentation across all phases of development	▪ Documented procedure for estimation of size and for using productivity measures ▪ Availability of data dictionary for controlling and storing details of all data files and their fields ▪ Coding standards: adherence to coding rules which emphasize communication via program code. ▪ Detailed documentation, including project plans, models, requirements, code, test cases	Rational Unified Process (RUP)	Structured methods
	▪ Design documentation is minimized or generated automatically by the programming environment used to implement the system ▪ (R) Documentation via oral communication, the code itself, and tacit knowledge transfer	XP	Agile
	▪ (R) Working software in replace of documentation ▪ (R) Primary permanent artifacts that should be invested in for a software system are the code and automated tests. ▪ (R) Executable test cases		Agile
	▪ Specifying deliverable documents	Military standards	
	▪ The use of documentation standards helps to ensure that proposals are complete and that they are communicated to bother users and developers.		Structured Method
Defect detection	▪ Rigorous testing (vs. ad hoc testing)	Pragmatic programming	
	▪ Recoding of vital defects in the problem tracking systems ▪ Customers write function tests to demonstrate the features implemented.	XP	Agile

Defect prevention	▪ Collection of statistics on sources of errors and their analysis for cause detection and avoidance	SBPQ	
	▪ Gathering statistics of testing efficiency and their analysis	SBPQ	
	▪ Testing through life cycle. Testing is done at the end of each iteration and for each new release of the project.	RUP	
	▪ Test planning prior to programming	SBPQ	
	▪ Test-first programming: one should write a test before writing code, and all code undergoes exhaustive testing as soon as it is written.	XP	Agile
	▪ Programmers continually write tests which must be run flawlessly for development to proceed.	XP	Agile
	▪ Automation-driven root cause analysis of failures with a focus on finding the underlying reason a failure has been found by a tester or has been experienced by a user.	XP	Agile
	▪ Adoption of design patterns		
	▪ The development and implementation of a defect prevention plan		
	▪ Acceptance test-driven development (a practice whereby acceptance tests are written as a collaborative effort between the product manager/customer, tester, developer, and user interface designer early in the iteration).		Agile
	▪ Setting up a system architecture that supports regression testing (automated batch testing) and to build test cases as they go	Crystal SBPQ	
	▪ Collection of statistics on sources of errors and their analysis for cause detection and avoidance	SBPQ	
	▪ Pair programming: All production code is written by two programmers at one machine.	XP	Agile
	▪ Automated acceptance test	Test-Driven Development	

		(TDD)	
	▪ Automated unit tests	TDD, XP	
	▪ Test in a clone of the production environment		
	▪ Designing quality and integrity into the product (Cobb, 2011)		Agile, Lean
	▪ Evaluate all defects for correction and to identify, fix, and prevent other similar problems	Humphrey and Thomas (2010)	
	▪ Establish and maintain a requirement quality-management process	Humphrey and Thomas (2010)	
	▪ Establish quality policies, goals, and plans	Humphrey and Thomas (2010)	
	▪ Use statistical process control and mathematically based verification to develop software with certified reliability	Cleanroom	
Regular inspection and review of deliverables	▪ Independent audits (walkthroughs and inspections of design and code) conducted at each major stage	SBPQ	
	▪ Formal procedure (like review or handover) with sign-off for passing over deliverables from one group to another	SBPQ	
	▪ User reviews on a regular basis (short rather than long interval, such as quarterly reviews)		
	▪ Peer reviews		CMMI
	▪ Review, inspect, and evaluate all product artifacts	Humphrey and Thomas (2010)	
	▪ Sequential reviews ad audits	Military standards	
	▪ Verify software quality continuously	RUP	
	▪ Sprint Review at the end of each iteration, including demonstrating features to the customer, management, users, and the product owner; and review the project from a technical	Scrum	Agile

	perspective		
Modular design (or architecture-based design)	▪ Function decomposition		
	▪ The use of components and a modular architecture extends concepts of modularity and encapsulation from object-oriented design	RUP	
	▪ Use component-based architecture (architecture definition using components)	RUP	
Plan-driven design	▪ Careful upfront and extensive design and analysis	Feature-Driven Development (FDD), Object-oriented methodology, the Project Management Institute's PMBOK, PRojects IN Controlled Environments (PRINCE2)	
	▪ Testing after most of works have been done		
	▪ Freeze requirement after small number of iterations		Structured methods
	▪ Substantial attention to architecture	RUP	
Project knowledge management	▪ Code is owned by a particular developer, often based upon a developer's area of expertise		
	▪ The importance of tacit knowledge: agility is achieved by establishing and		Agile

	updating project knowledge in the participants' heads rather than in documents		
	▪ Pair programming rotations	XP	
Substantial attention to requirements analysis	▪ A rigorous requirements management approach to ensure that all business rules are captured and integrated into the design of application (Cobb, 2011)		
	▪ Creating an acceptable rich picture to stimulate debate and ensure richer understanding of the problem situation that achieved by conventional hard analysis techniques.	Soft Systems Methodology (SSM)	
	▪ Use cases to capture functional requirements	RUP	
	▪ Joint Application Design (JAD) and Joint Requirements Planning (JRP) workshops are extremely intensive session of short duration which serve to identify more completely problematic requirements analysis and system design issue.		RAD
	▪ Understand users' business processes, have done the necessary business process analysis and rationalization, and understand how automation might change their process		
	▪ Manage requirements	RUP	
Explicit recognition of risk	▪ Risk management during ISD, including risk identification, analysis, planning, and monitoring	Spiral	
	▪ Throwaway prototyping		
	▪ Risk, such as design flaws, failing to meet user needs, escalating costs, losing sight of the perceived system benefits, etc., are detected early in the development process	Spiral	
	▪ Risk-driven development (the hardest, riskiest problems first)	UP/RUP	
	▪ Formal assessment of risk, benefits, and viability of projects prior to contractual commitment	SBPQ	

	▪ Case-based reasoning ▪ Scenario planning ▪ Future Analysis ▪ SWOT analysis		
	▪ Synergistic contingency evaluation and review technique	Cooper and Chapman, 1987	
Conscious efforts to make project size, cost, and schedule estimation	▪ Cost estimation, such as the Constructive Cost Model (COCOMO)		
	▪ Feasibility analysis ▪ Function point analysis ▪ Work breakdown structure		
	▪ Fix time and resources, and then to adjust the amount of functionality accordingly (i.e., a timeboxing technique)	Dynamic systems development method (DSDM)	RAD
	▪ Formal procedure for estimation of effort, schedule and cost	SBPQ	
	▪ Procedures to ensure that the functionality, strengths, and weaknesses of the system which the software is replacing are reviewed	SBPQ	
	▪ Trade-off analysis	Renaissance	ESPRIT
	▪ Planning is based on dependencies, risk, complexities, workload balancing, and client-required milestones	FDD	Agile
	▪ Planning Poker technique for estimation and planning (based on expert opinions and team consensus)	Wideband Delphi Estimation, Scrum	Agile
	▪ Estimation based upon the notion of story points (unit-less measures of effort relative to previously completed requirements)		Agile
	▪ (R) Working software is the principal measure of progress		Agile
Tool/Software support	▪ Manage changes to the software using a	FDD	Agile

	change management system and configuration management procedures and tools		
	▪ Maintaining awareness of CASE or other new software engineering technologies	SBPQ	
	▪ Usage of tools for tracing forwards and backwards through requirements, design, and code	SBPQ	
	▪ Usage of software tools for tracking and reporting the status of the software/sub-routines in the development library		
	▪ Automated methods to enable the standardization (Cobb, 2011)		Lean
	▪ UML	Object-oriented Methodology & RUP	
	▪ Model software visually	RUP	
	▪ The use of integrated CASE wherever possible		RAD
	▪ Tools, such as visual modeling tools, are used to support a development methodology	RUP	
Project monitoring and tracking	▪ Project monitoring and review		Project management
	▪ Frequent management activities aiming at consistently identifying any deficiencies or impediments in the development process as well as the in the practices that are used	Scrum	Agile
	▪ Record and feedback of estimated versus actual efforts into estimation process	SBPQ	
	▪ Maintenance of records and feedback of size into estimation process	SBPQ	
	▪ Using of “earned value” project tracking to monitor project progress	SBPQ	Project management
	▪ Production of estimates, schedules, and changes only by the project managers who directly control the project resources	SBPQ	

	▪ Daily scrum meetings (synonyms: stand-up meetings)	Scrum	Agile
Management of sponsors and champions	▪ Seeking top management support, including communication of objectives, resources commitment, etc. ▪ Managing expectations		
Close control over software development processes and procedures	▪ Obtaining signoff from all parties before changing plans by the business project manager	SBPQ	
	▪ One or more documents should be approved after each phase ▪ Procedures for controlling changes to requirements, design, and documentation ▪ Procedures for controlling changes to code and specifications		Structured methods
	▪ Procedure to check that the system configuration passing user acceptance is the same as that implemented	SBPQ	
	▪ Gathering statistics of testing efficiency and their analysis	SBPQ	
	▪ Code cannot be considered "done" until it passes the set of done criteria established by a team at the start of the release Some examples of done criteria include the following: passing all acceptance tests, 80% unit test coverage, no high severity defects are open for the future. The feature must be moved into the next iteration		
	▪ Repeatable stage: the organization has achieved a stable process with a repeatable level of statistical control by initiating rigorous project management processes to track cost, schedule, and functionality.	CMMI	
	▪ Defined Stage: the software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization.	CMMI	

	Control change to software (isolated changes and using procedures)	RUP	
	Time boxing: the subset of the requirements being developed during each one month iteration is frozen into the "iteration backlog" at the beginning of the iteration.	Scrum	Agile
Management and control via metrics	All results are quantifiable		
	Both the software process and product are quantitatively understood and controlled		CMMI
	Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.		CMMI
	Plan leading indicators and metrics to manage the project (Cobb, 2011)		
	To know how to improve the quality of personal work, developers need objective data (the defect they personally inject and remove, the size of their products, and the time they spend)	Humphrey and Thomas (2010)	
	Measure and use inspection data both to improve the inspection process and to focus the inspection on the most defective product elements	Humphrey and Thomas (2010)	
	Establish and maintain statistical control of the software engineering process	Humphrey and Thomas (2010)	
	Provide precise measures for teams	TSPi (Team Software Process)	
	Establish standard measures for quality and performance	TSPi	
Collective ownership	All code is collectively owned. All programmers should know enough about the system to make any necessary changes to improve code.	XP	Agile
	(R) Code is owned by a particular developers, often based upon a developer's area of expertise	FDD	Agile

Promote simplicity	▪ Producing a product that is simple enough to handle change while fulfilling customer requirements	XP	Agile
	▪ Working software is the principal measure of progress		Agile
	▪ Eliminating waste, such as untested code, a partial implementation of a customer requirement, and extra features (if your design has capabilities that are beyond the current user stories or that anticipate new features, you should expend extra effort to remove them)		Lean development
	▪ Use short statements to describe the functionality desired by a system user or customer to communicate rather than documenting the details for requirements that may never be chosen due to changing requirements, competition, and changing environment (e.g., story cards)	FDD, XP	Agile
	▪ Differentiate what clients want from what clients need by keep asking why they want		
	▪ "Just Barely Good Enough" requirements	XP	Agile
	▪ Drive out waste through standardization, process standardization, and skill-set standardization (Cobb, 2011)		Lean
Project transparency	▪ Big, visible charts that provide updated status information such that it is worth the team member's time to look at the display. Different forms of status information can be presented through Task Kanban board, the burn down chart, and the iteration status board.	Kanban, FDD	Agile
	▪ Visible release and iteration backlog		
	▪ Teams can provide Intranet- or Internet-based informative workspaces, such as team-based task tracking software.		
	▪ Reporting/visibility of status	FDD	Agile

	▪ Make progress visible to all (Cobb, 2011)		Lean
Reflect on improvement at regular intervals	▪ Retrospective meetings		Agile
	▪ Kaizen improvement: continuous evaluation, incremental change, and improvement through tuning	Renaissance	ESPRIT
	▪ Use lessons learned from past projects for future projects (Cobb, 2011)		Lean
	▪ Continuously improve the development process (Humphrey & Thomas, 2010; Cobb, 2011)		Lean
	▪ Optimizing stage: continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.		CMMI
Leverage industry standards or solutions for IS projects	▪ UML: an industry standard graphical language for describing, constructing, and documenting system components (artifacts).	OO	
Developing capabilities of ISD professionals	▪ Training	People-CMM	
	▪ The identification of core competencies developed by the organization	People-CMM	
	▪ Having individuals being able to establish their own program of professional development	People-CMM	
	▪ Developing capability of finding, understanding, and adapting reusable components	Component-based development	
	▪ Nurture a learning environment (Cobb, 2011)		Lean
Planning and shaping the workforce	▪ Establishing basic staffing practices	People-CMM	
	▪ Developing plans for workforce development	People-CMM	
	▪ Sets and tracks objectives for competencies in the workforce	People-CMM	
	▪ At least one experienced person on the team	Crystal	

	▪ Having good people and strong domain experts	FDD	Agile
	▪ Building an organization based on respect for people - treat people as most valued assets (Cobb, 2011)		Lean
	▪ The team aspect serves to ensure that all the vital skills for successful development are present. Successful development requires that many varied roles be accommodated. The fulfillment of these roles is facilitated by team composition, and their importance has been acknowledged by Microsoft who organize their software development around various development roles in a similar manner	RAD	
Motivating and managing performance	▪ Establishing basic performance management and compensation practices	People-CMM	
	▪ Improving performance management and compensation practices through adaptation to competency development and team building	People-CMM	
	▪ Project should be built around motivated individuals, who should be trusted		Agile
	▪ A sustainable development that does not involve excessively long working hours and maintain a constant pace indefinitely (40-hour week)	XP	Agile
	▪ Scrum meetings – let team members to publicly commit to their work plans	Scrum	Agile
Team and culture building	▪ Establishing basic communication skill	People-CMM	
	▪ developing a participatory culture	People-CMM	
	▪ Formal team-building and continuous improvement of team capabilities	People-CMM	
	▪ The engendering of a non-blame culture and all pulling together		
	▪ Use self-organizing teams		Agile
	▪ At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly		Agile

	▪ Participants that traditionally belong to separate teams (e.g., testers and designers), are integrated into the development team and process.		Agile
	▪ As a team grows in capacity (due to experience), keep their workload constant but gradually reduce the size of the team	XP	Agile
	▪ Keep effective teams together	XP	Agile
	▪ The cross-functional team of all those necessary for the product to succeed as one team		
	▪ The team strives for joint objectives		
	▪ Building a self-organizing team to enable best architectures, requirements, and design.		
Empowerment	▪ Open for the developers to choose the specific software development techniques, methods, and practices for the implementation process	Scrum, Crystal	
	▪ Small empowered teams. Teams are empowered to make vital design decisions.		RAD
End users' welfare is the major concern of ISD	▪ Ethnographic approaches to understand how IS can be fitted into work lives of the organization using it	Multiview 2	SSM
	▪ Analysis and design of socio-technical aspects, such as analyzing how the system will affect users and their jobs, how a system can be fitted into working lives of the people in the organization sing it.	Multiview	
	▪ Diagnosis of job satisfaction needs	Effective Technical and Human Implementation of Computer-based Systems (ETHICS)	Socio-technical
Frequent releases to	▪ Rapid development cycles (synonym: Iterations, Sprint) (Crystal Clear	XP, Evolutionary,	Agile, RAD

customers	requires deliveries to be no longer than three months at worst).	Spiral, Crystal Clear Scrum	
	▪ Frequent delivery of products which focus primarily on satisfaction of business functionality. Shorter time-box for development – typically 90 days – are an important feature. These shorter time-boxes make project management more straightforward in that it is easier to focus on necessary activities and be more accurate as to what can be achieved.		RAD
	▪ Deliver working software frequently		Agile
	▪ Frequent executable release	RUP	
	▪ Short iterations		
	▪ Forcing implementation of only the highest priority functions		Agile
Identify IT/business strategy and align projects with IT/business strategy (project alignment)	▪ Adoption of the Zachman framework		
	▪ Welcoming changing requirements		Agile
	▪ There needs to be an effective linkage between business strategy and business processes. The business-oriented process vision can developed through: assess existing strategy with respect to processes, consult with process customers for performance objectives, and set up performance objectives and functionality targets.	Process Innovation	
	▪ Scenario planning		
	▪ Future Analysis		
	▪ SWOT analysis		

	▪ Implement requirements traceability to ensure that the requirements are complete, sufficient, and consistent with the business objectives that they are intended to fulfill (Cobb, 2011)		
	▪ An overall organizational strategic orientation is necessary in system development (strategic modeling, tactical modeling, and operational modeling)	Information Engineering	
	▪ Analyzing how a system is supposed to further the aims of the organization installing it	Multiview	SSM
Continuous attention to customer problems and satisfaction	▪ Business domain analysis		
	▪ An inclusion of business experts		
	▪ Giving customers control over features and priorities		
	▪ End-of-iteration customer focus group	RADical Software Development	Agile
	▪ Functional decomposition which breaks down a "business activity" requested by the customer to the features that need to be implemented in the software.	FDD	Agile
	▪ Features are prioritized by high business value and risk	Scrum	Agile
	▪ Place the highest priority on customer satisfaction achieved through early and continuous development		Agile
	▪ Establish the value of the end product or system to the customer (Cobb, 2011)		Lean
	▪ Frequently involve the customer (Cobb, 2011)		Lean
	▪ Use a defined process for capturing requirements focused on customer value (Cobb, 2011)		Lean
	▪ Understand the quality characteristics that are most important to the users and to measure these characteristics in a	Humphrey and Thomas (2010)	

	way that is meaningful both to you and to the users

Appendix B

Elicitation Techniques[27]

Elicitation Techniques	Description[28]	Evaluation
Cognitive interviewing techniques	■ Elicitation: several cognitive interviewing techniques (e.g., open interviews, question-answer interviews, inferential flow analysis) can be used to elicit mental models ■Analysis: a transcript of the interview is constructed and analyzed using propositional or discourse analysis ■ Representation: a graph that illustrates domain concepts and conditional and causal associations among them.	■ a starting point for obtaining information about the domain of interest or for investigating individual mental models in loosely structured domains. (-) it relies heavily on the interviewer's interpretations (-) interview techniques capture only information that can be expressed verbally (-) interviews conducted after task performance can be subject to retrospective distortions. (+) it can be used to elicit a team mental model directly through group discussion (caveats: dominant opinions and facilitator's skills)
Verbal protocol analysis	■ Elicitation: participants are asked to think aloud while they undertake a task or make a decision. Sessions are recorded on audiotape or videotape, and a written protocol is generated later. ■ Analysis: the researcher can	■ appropriate for researching tasks for which verbalization is a natural part of the thinking process ■ appropriate for use in small-scale mental model research in domains in which verbalization is a normal part of task performance (e.g., decision making). (-) the labor-intensive process of collecting, scoring, and analyzing protocol data can limit its use

[27] The table is adapted from Langan-Fox et al. (2000), Mohammed et al. (2000), and Mohammed et al. (2010).
[28] Elicitation refers to the technique used to determine the components or content of a mental model. Representation refers to the technique used to reveal the structure of data or determine the relationships between elements in an individual's mind (Mohammed et al., 2000).

	identify the relationships between objects within a domain. ■ Representation: sets of production rules, decision trees, heuristics, algorithms, systematic grammar networks, and means-ends hierarchies	(-) verbal accounts can be incomplete or erroneous, given that people do not always have conscious access to the thought structures that underlie their behavior and that people's beliefs are not always concordant with their actions (Johnson & Stephens, 1994) (-) experimental situation itself might encourage participants to construct a rationale for their behavior that might not otherwise exist. (-) the individual-level output produced by verbal protocol analysis might be difficult to summarize and compare systematically, which limits the usefulness of the technique for team mental model measurement.
Content analysis	■ Elicitation: content analysis is a systematic method for analyzing written statements such as formal speeches (Hart, 1977) and transcripts of interviews (Langan-Fox &Tan, 1997). ■ Analysis: the researcher uses a set of coding rules to analyze sentences phrase by phrase to uncover important concepts and the relationships between them ■ Representation: these relationships can be represented graphically (Wrightson, 1976)	■ the use of content analysis only when there is no possibility of eliciting the mental model from the participant in person (+) it is not constrained by the availability of respondents (-) there is no guarantee that a written statement is a faithful representation of an individual's beliefs. (-) the mental model generated by the researcher might be erroneous or incomplete because the individual is not present to endorse the existence of certain beliefs or clarify contradictions. (-) the validity of content analysis for deriving a team mental model is doubtful; there is no way to assess the accuracy of this representation in projecting the views of each individual in the group.
Observation of task performance.	■ Elicitation: researchers can use direct observation of an individual's behavior during the completion of a task to infer mental models	■ observation of task performance is best suited to the examination of (individual) mental models in contexts for which user-system interaction is highly structured, and we recommend its use when error detection is of

		particular interest. (-) it would be difficult to determine shared understandings of a domain through observing behavior. (-) the assumption that individuals have conscious access to the cognitive structures that underlie their behavior
Visual card sorting	■ Elicitation: the participant is either provided with researcher-generated concepts or is asked to list all the concepts that he or she sees as relevant to the domain of interest. The concepts are written on cards, and the participant is asked to sort the cards on a surface by placing cards that are perceived to be related closer together. The participant then explains why he or she arranged the cards in such a way. This information is tape recorded or transcribed, and the arrangement of cards (the final representation) is photographed.	■ the use of visual card sorting when research time is restricted but caution that care should be taken when using the technique for team mental model measurement. (Δ) represent knowledge that is easily and often accessed from short-term memory (Δ) the visual card sorting technique can be used in a group session to measure the team mental model (caveats: dominant opinions) (-) visual card sorting elicit placement judgment only without any quantitative indices (Mohammed et al., 2000)
Repertory grid technique	■ Theoretical basis: personal construct theory (Kelly, 1955) ■ Elicitation: elements and concepts are elicited in an interview (or provided by the researcher). Perceptions of elements and concepts can be collected by sorting, ranking, or rating. See	(+) high validity (it is well grounded theoretically) and reliability (it produces similar representations over time) (+) the technique is suitable for use with participants at various levels of cognitive ability. (-) the prohibitive amount of time required to administer the technique (Δ) RGT does not provide a direct method for eliciting team mental models, and it is unlikely that individual

	details in the research methodology section. ■ Analysis: both qualitative and statistical analyses can be performed on individual grids to determine a participant's pattern of dimensions and knowledge structure (Langan-Fox &Tan,1997). E.g., content analysis, rearranging the RepGrid, decomposing the RepGrid, analyzing content and structure (Simpson and Wilson (1999): MDS, correlation, and cluster analyses; Daniels (1994): Cochran Q Test, Helmert contrasts, and ANOVA).	models can be used to derive a team map indirectly.
Causal mapping	■ Elicitation: the participant is asked whether one concept influences the other, if it does so positively or negatively, and whether it does so weakly, moderately, or strongly for each possible pair of a set of concepts (Concepts can be generated by the researcher or by the participant). In addition to interactively requesting the data from participants through questionnaires and/or interviews, causal map can be elicited from post hoc analyses of data (e.g., systematic coding of documents or transcripts).	(+) according to Gray, Bougon, and Donnellon (1985), "causality is conceptually and instrumentally the most potent of all relations" (p. 85).

	■ Analysis: A distance ratio formula can be used to infer the extent of difference between the maps of individual team members.	
Pairwise rating methods	■ Theoretical basis: associative memory theory ■ Elicitation: the researcher collects similarity or relatedness ratings (proximity ratings) on each possible pair of concepts from the total concept pool. These ratings are usually obtained by asking the participant (for each pair), "How similar or related are A and B?" on some sort of scale - for instance, not at all related (1) to highly related (9). ■ Analysis: multidimensional scaling and general weighted networks such as Pathfinder	(+) it is time efficient (120 ratings can be completed within 30 min) (+) it requires little reading or writing (concept pairs can be read out to participants) (+) it is indirect (the mental model is not articulated by the participant but, rather, inferred through statistical analyses). (+) high validity and test/retest reliability (-) the repetitive nature of pairwise ratings can induce a response set. (-)the participant doesn't specify their definition of similarity ((e.g., causation, co-occurrence, dependency, contingency) (Mohammed et al., 2000) ■ use when research time is constrained or when the English skill or education of the participant sample is limited.
Ordered tree technique	■ Elicitation: participants are asked to recall a large, well-learned set of items many times from many different starting points[29] -Analysis: recall strings are analyzed using an algorithm that produces a tree with directional nodes indicating the order of	(+) it provides information about the amount of organization in a cognitive structure (i.e., the degree to which concepts are separated or clustered), the depth of the structure (i.e., the degree to which concepts are related to higher-order, more general concepts), structure similarity (i.e., the number of nontrivial chunks that two structures have in common), and the order of information in the structure (i.e., consistencies

[29] For example, in free recall, the participant traverses an ordered tree beginning at the root and descends the nodes until a terminal item is reached. Upon recalling that item, he or she moves up to its immediately superior node and descends the constituent link until all of its descendent terminal items have been recalled.

	traversal.	in order in nondirectional, bidirectional, and unidirectional chunks). (-) the technique is limited by its reliance on retrieval processes.

Note: (+) means pros, (-) means cons, and (Δ) means neutral.

Appendix C

Analysis and Representation Techniques[30]

Analysis and Representation Techniques	Description	Evaluation
Multidimensional scaling (MDS)	■ MDS generates spatial configurations that give a pictorial representation of how concepts are clustered within that space. ■ Assumptions: concepts that possess common features or characteristics should be rated as similar and should be located closer together in space. ■Applications: (1) the technique can be used to identify the dimensions that an individual uses to judge the similarity (or relatedness) between clusters of concepts and the dominance of a particular concept within an individual's mental model. (2) MDS can be used to draw comparisons among the mental models of different individuals and has potential for team mental model measurement.	(+) the strength of a dimension in a mental model is calculated in terms of a structural ratio (i.e., the ratio of the mean distance between concepts in the same category to the mean distance between concepts in different categories). (+) MDS is particularly useful in identifying the unknown underlying dimensions used to cognitively Organize stimuli (Kruskal & Wish, 1978; Nunnally, 1978). (Δ) MDS is problematic in that there are a number of variations of scaling techniques to choose from and the most appropriate technique is not always easy to identify. (-) MDS does not present links between concepts. Rather, concepts are represented in n-dimensional space.
Distance ratio formula (DR)	■ DR calculates the degree of similarity between two maps, represented as expanded association matrices. The general idea of the	(+) it can be used to isolate three types of differences: differences in the strengths of commonly held beliefs, differences

[30] The table is summarized from Langan-Fox et al. (2000), Mohammed et al. (2000), and Mohammed et al. (2010).

	formula is to sum the differences between two maps and then divide that sum by the greatest possible difference, given the number of concepts in each map and the number of concepts common to the maps. ■ Applications: This formula can be used to calculate the average degree of overlap between the mental models of each member of a team by calculating the DR for each possible pair of team members, summing the DR values, and taking the mean.	attributable to the existence or nonexistence of beliefs involving common concepts, and differences attributable to beliefs consisting of unique concepts. (-) the formula treats the absence of a link between two concepts the same as the absence of a link attributable to the absence of a concept. (-) the formula cannot be generalized to maps of different types.
Pathfinder	■ Pathfinder is a computerized networking technique that is used to derive associative networks based on perceived relatedness among a selected set of concepts. As psychological models, networks assume that concepts and their relationships can be represented by a structure consisting of nodes and links. ■ The PF algorithm transforms raw, paired comparison ratings into a network structure in which the concepts are represented as nodes and the relatedness of concepts are represented as links between nodes (Schvaneveldt, Durso, &Dearholt, 1989; Schvaneveldt, 1990). ■ Formula: the path length, Pathfinder network similarity, Pathfinder average ■ Applications: (1) the Pathfinder network similarity (NETSIM) function can be used to reveal	(-) the layout of items in a Pathfinder network is arbitrary (i.e., it represents associative but not semantic information about conceptual relationships). (+) it reveals cognitive structure. The output of PF is a network representation (PFNET) in which relatedness between concepts is depicted by how closely they are linked, and weights represent the strength of the links (Schvaneveldt et al., 1989)

	differences in the way knowledge is structured in two different networks (i.e., to examine the extent to which there is a shared mental model). The average NETSIM of a team can be determined by averaging the sum of each possible pair of NETSIM values within a team. (2) the Pathfinder Average function can be used to construct an average of a network that combines the proximity ratings data for all the members of a team to produce an aggregate network. (3) other measures: Coherence measures, Closeness measures, the number of links (indirect measure of knowledge structure complexity), distance between nodes in the network, levels of link strength, starness (used to determine levels of abstraction), and the z statistic (to produce composite networks)	

Note: (+) means pros, (-) means cons, and (Δ) means neutral.

Appendix D

Results of Repertory Grids

Participant #1

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB[31]	Different (9)	Categorization
traditional system analysis and design that focuses on system requirements	9	3	5	7	7	1	1	1	stakeholder analysis to determine the stakes and expectations of the stakeholders and communicate/negotiate better with the stakeholders	Management of sponsors and champions
Leverage external expertise for complicated technical issues	1	7	3	3	1	6	2	9	Solve complicated technical issues by internal knowledge	Workforce planning
less disciplined project management -a lack of project monitoring	9	1	5	7	7	2	1	1	formal project management -strict monitoring and project control -risk assessment	Close control over software development processes and procedures, Project monitoring and tracking , Explicit recognition and management of risk
evolving development	6	9	7	5	5	8	8	9	Upfront thorough system analysis and design and the freeze	Plan-driven development, Iterative and incremental

[31] SUB means suboptimal (unsuccessful).

processes									requirements	development
strong vendor support	5	9	1	3	1	1	3	9	strong community support (e.g., open source community)	Close collaboration and communication between project members during development
Get buy-in using business cases	2	9	4	2	1	8	3	9	no buy-in from stakeholders	Management of sponsors and champions
intensive user involvement -users work closely with developers and help testing -users involve throughout the whole project cycle	1	7	5	5	3	9	9	9	less user involvement -users only participate in the initiation and closing stage.	User involvement and participation

Participant #2

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
iterative development	9	7	7	1	3	1	2	9	waterfall, sequential development	Iterative and incremental development, Plan-driven development
modular design	N/A	3	2	1	9	2	2	9	sequential or procedural design (tight module coupling)	Modular design

careful design for the overall system architecture (with detailed technical documents)	N/A	6	1	5	8	4	5	1 or 9	experimental and ad-hoc design (refactoring is employed)	Iterative and incremental development, Plan-driven development, Careful and comprehensive documentation across all phases of development , Innovative design, Continuous attention to technical excellence
a lack of documentation	8	2	9	8	1	7	7	1	adequate update on documentation (with the assistance of automatic document generation)	Careful and comprehensive documentation across all phases of development , Tool/Software support
a lack of version control and source code management	9	3	9	9	1	9	9	1	version control	Close control over software development processes and procedures, Tool/Software support
a lack of control for programming	7	1	7	8	1	8	7	1	setting programming standard and establishing norms of good programming	Standards for design
a lack of competent project members	7	2	8	9	1	8	5	1	good team composition (involving senior developers and looking for competent developers)	Workforce planning

Participant #3

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
close collaboration -open floor -stand-up meetings	1	1	3	1	1	1	5	9	dispersed collaboration	Close collaboration and communication between project members during development

iterative analysis and design	3	2	6	2	6	4	1	9	structural analysis and design, such as rational unified process and OOAD	Iterative and incremental development, Plan-driven development
strong knowledge/expert support	1	1	2	1	2	7	2	9	absence of support	Workforce planning
resolve escalated issues	N/A	N/A	N/A	1	3	1	2	9	unclear escalating process	Effective escalation management process
incremental release (quick release)	9	9	9	5	9	5	1	N/A	full feature release	Frequent releases to customers, Iterative and incremental development
version/source code control	1	1	1		1	1	1		a lack of version/source code control	Close control over software development processes and procedures, Tool/Software support
scalable human resource management	9	9	9	9	9	7	1	N/A	dedicated human resources	Workforce planning
idea generation from multiple experts	9	9	9	1	5	9	1	N/A	dedicated human resources	Workforce planning, Innovative design
vendor-owned development & production environment	9	9	9	5	1	5	1	1	client-owned development & product environment	Standards for design
cloud-based development environment	9	9	9	1	9	9	1	9	local (internal) network development environment	Standards for design
rigorous quality control through testing	1	1	1	1	3	7	4	9	a lack of quality control	Defect detection

Participant #4

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization

informal specifications, such as emails and chat	1	4	7	7	9	8	6	1 or 9	well-defined specification (with too much details)	Careful and comprehensive documentation across all phases of development
infrequent feedback from product owners	3	1	7	9	7	5	7	1	frequent feedback from product owners	User involvement and participation
infrequent release to the production environment	1	4	8	9	4	4	8	1	frequent release to the production environment	Frequent releases to customers
a lack of continuous integration	1	3	6	9	8	8	8	1	continuous integration	Continuous integration
manual unit testing	1	1	7	9	7	8	6	1	automatic unit testing	Tool/Software support, Defect detection
manual integration	1	1	6	9	8	8	8	1	automatic integration	Continuous integration, Tool/Software support
infrequent release to the testing environment	3	4	8	9	8	9	8	1	frequent release to the testing environment	Continuous integration, Defect detection
task estimation based on team decisions	9	8	4	1	3	1	2	9	task estimation based on individual decisions	Close collaboration and communication between project members during development, Conscious efforts to make project size, cost, and schedule estimation
task estimation based on story points	9	9	1	1	1	1	9	9	task estimation based on time	Conscious efforts to make project size, cost, and schedule estimation
test-driven development	9	9	1	1	2	1	9	9	non test-driven development	Defect prevention, Continuous attention to technical excellence
pair programming for better software quality	9	9	7	3	2	1	5	9	individual programming	Defect prevention

team-oriented decision-making	7	9	5	4	4	2	2	1 or 9	**dictatorial** decision-making	Close collaboration and communication between project members during development, Empowerment
project rotation	9	2	5	5	2	6	2	9	no rotation	Workforce planning, Develop capabilities of ISD professionals

Participant #5

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
linear development (plan-driven)	2	4	8	8	6	7	8	1	iterative and adaptive development	Iterative and incremental development, Plan-driven development
collective ownership of project, such as visible status, report, and outcomes	3	7	1	1	2	2	5	9	a lack of knowledge(awareness) of overall status and outcomes	Collective ownership for development processes and outcomes, Project transparency
automated testing	2	4	8	1	N/A	7	2	9	a lack of automated testing (manual testing)	Continuous integration, Defect detection, Tool/Software support
documentation for approval(mandatory and may not be useful)	2	4	9	7	8	9	9	1	documentation based on needs (useful)	Promote simplicity
small releases for demonstration, including working skeleton, prototyping	3	8	3	1	2	9	3	9	big bang releases (until the end of the project) -higher risk	Iterative and incremental development, Frequent releases to customers

end-user collaboration (with intensive involvement)	2	6	6	1	2	8	3	9	a lack of end-user collaboration (speculation of requirements and issues from developer teams)	User involvement and participation
collaborative decision-making (consensus-based)	3	5	1	3	3	2	8	9	dictatorial decision-making	Empowerment, Close collaboration and communication between project members during development

Participant #6

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
a lack of quality assurance management (e.g., developers conducted testing on their own)	8	1	1	7	1	1	1	1	systematic quality assurance management (e.g., QA role, QA plan, QA procedures)	Close control over software development processes and procedures, Defect prevention, Workforce planning
Automated testing	7	2	3	2	1	2	3	6	Manual testing	Defect detection, Tool/Software support
on-site customers	5	3	2	1	2	3	3	9	No access to users -rely on documentation and specification	User involvement and participation
iterative prototyping for learning and innovation	7	2	2	1	7	4	5	9	heavy up-front design, plan-driven design	Iterative and incremental development, Innovative design, Plan-driven development
project and domain knowledge repository and transfer mechanisms, such as document repository, show and tell, and learning sessions	1	6	7	5	5	7	6	7	a lack of knowledge repository and transfer	Project knowledge management

Personal skill development	2	3	5	1	6	4	7	9	Personal skill development isn't a focus	Develop capabilities of IS development professionals
Modular design	5	2	7	1	8	6	8	9	integrated (or coupled) design	Modular design
rigorous progress tracking -time tracking systems -visible progress chart	2	3	7	1	5	7	6	9	progress invisible to team members (only project managers are aware of progress)	Project monitoring and tracking , Project transparency
new request management	4	3	5	3	2	3	6	7	a lack of request management	Disciplined change evaluation and management

Participant #7

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
shorter development cycle for frequent feedback	6	4	1	2	2	2	6	8	longer development cycle	Frequent releases to customers, Iterative and incremental development
single thread version control (CVS)	1	1	9	7	9	1	1	1	multithread version control (Mercurial) (management of multiple versions in parallel)	Close control over software development processes and procedures, Tool/Software support
collaborative design with on-site customers (frequent communication with users)	9	2	1	2	5	8	6	9	design based on restricted requirement (early freezed documentation guide development)	User involvement and participation, Plan-driven development
"hands-on everything" project management (close control)	4	2	2	2	1	8	4	8	loose control	Empowerment
project management following the structures	3	4	4	4	8	1	3	1 or 9	improvised project management (chaos management)	Iterative and incremental development, Plan-driven development

Participant #8

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
Frequent project review(e.g., status report)	1	8	2	3	5	5	9	7	Infrequent project review	Project monitoring and tracking
QAs have sophisticated domain and application knowledge	9	3	1	4	9	5	9	9	QAs have testing knowledge and techniques alone	Workforce planning
Software effort estimation based on few key players	1	9	1	4	3	3	3	9	Software effort estimation based on negotiation and consensus	Conscious efforts to make project size, cost, and schedule estimation
limited reusability	7	7	9	7	2	5	9	1	high reusability design (component-based design)	Modular design, Reuse of code and components
responsive resource allocation	1	3	7	4	2	4	5	9	a lack of responsive resource allocation	Management of sponsors and champions
pair programming	1	4	6	4	7	8	9	9	linear development	Close collaboration and communication between project members during development, Continuous attention to customer problems and satisfaction, Defect prevention
code review	2	7	3	3	9	4	9	9	a lack of code review	Defect prevention, Regular inspection and review of deliverables, Continuous attention to technical excellence

project managers employ coordination and communication skills to resolve project issues that come from stakeholders	1	2	9	2	6	3	6	9	project managers do not have appropriate capability to deal with project issues (project managers receive issues from stakeholders and pass them to subordinates. They do not proactively solve problems)	Management of sponsors and champions , Workforce planning

Participant #9

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
comprehensive requirement documentation at the design phase	1	2	7	1,9	9	8	6	5	comprehensive documentation at the implementation phase	Careful and comprehensive documentation across all phases of development
minimum business commitment	4	2	8	9	8	9	9	1	maximum business commitment	Management of sponsors and champions
extremely detailed and full-scale testing	1	5	6	3	8	7	7	NA	high-level testing: functioning software is a primary goal	Defect detection, Promote simplicity
overarching enterprise architecture that documents business/IT plan, strategy, and roadmap	6	3	3	1	2	2	2	9	a lack of overarching enterprise architecture	Identify IT/business strategy and align projects with IT/business strategy
overarching IT governance process in order to align business/IT plan, strategy, and roadmap	6	4	3	1	1	1	2	9	a lack of IT governance process	Identify IT/business strategy and align projects with IT/business strategy

open and transparent communication with customers on IT solutions	5	4	3	1	3	6	6	9	no communication with customers on IT solutions	User involvement and participation
adoption of industry standard (e.g., ITIL, PMBOK, COBIT…)	4	4	3	1	2	5	4	9	no project-based standard	Leverage industry standards or best practices for IS projects

Participant #10

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
requirements and design specification are comprehensively and accurately documented	3	4	5	2	5	3	4	6	requirements and design decisions are communicated and exchanged through informal meetings	Careful and comprehensive documentation across all phases of development
Effective and open communication with customers (the goal is to obtain accurate requirements)	3	5	2	1	3	7	4	7	Poor communication between customers and project teams. This causes misunderstanding of requirements and inaccurate expectation of deliverables	User involvement and participation
The scope of the product is clearly defined and communicated. Customers know the impact and cost of changes	5	4	3	1	3	7	5	8	Scope ambiguity causes problems in cost overrun and human resources management	Management of sponsors and champions , Well-defined project charter and project plan that project stakeholders can understand
flexible architecture	4	2	7	1	6	5	6	9	rigid architecture	Modular design
Coding standards	5	2	5	1	5	8	7	9	a lack of coding standards	Standards for design
accurate work breakdown structure	3	4	5	1	5	5	5	9	a lack of accurate project planning and estimation	Conscious efforts to make project size, cost, and schedule estimation

informal work relationships between project members. Project members know each other well.	2	4	3	1	4	7	4	9	formal work relationships between project members (minimal social interaction and a lack of understanding)	Team and culture building

Participant #11

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
clear business strategies and visions	3	7	2	2	2	4	1	6	a lack of business strategies, visions, and objectives	Identify IT/business strategy and align projects with IT/business strategy
§ share common languages/understanding between management and developers (or PM can act as a liaison)	3	8	1	3	2	3	1	9	a lack of communication between management and developers	Close collaboration and communication between project members during development, Management of sponsors and champions
systematic project monitor and review	3	7	4	3	3	5	1	9	causal project monitor and review	Project monitoring and tracking
involvement of technical lead/experts	3	4	7	3	8	4	1	6	a lack of attention to technical excellence	Continuous attention to technical excellence
dedicated human resources for each project	1	1	7	2	7	3	2	9	human resource pool	Workforce planning
ensure design "perfected" in the early stage of the project (with placement of abundant resources)	3	8	5	3	7	2	9	5	agile design based on customer feedback	Iterative and incremental development, Plan-driven development, User involvement and participation
a team is composed of experienced developers	3	3	5	1	7	4	1	9	a team is largely composed of	Plan the workforce, Workforce planning

									inexperienced developers	
Micro project management (dedicated project managers pay attention to details and use a hands-on approach)	5	8	7	4	9	9	2	1,9	macro project management	Empowerment
strictly follow the procedures and rules	3	8	1	3	2	7	3	7	flexible procedures and rules	Close control over software development processes and procedures
individual-based reward	3	2	5	7	8	1	6	5	team-based reward	Team and culture building, Motivating and managing performance
independent testers	8	7	9	3	9	1	3	9	developers as testers	Workforce planning
supportive leadership style	7	9	7	3	7	2	3	9	subordinates are criticized and reprimanded when they do something incorrectly	Team and culture building
appropriate buffer time estimation	8	9	7	7	7	2	6	1,9	a lack of buffer time estimation	Conscious efforts to make project size, cost, and schedule estimation, Plan-driven development

Participant #12

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
component-based estimation	1	1	4	4	4	7	1	1,9	function points analysis	Conscious efforts to make project size, cost, and schedule estimation

planned task assignment (jobs should be done by specific person and time according to the plan)	9	1	3	2	1	5	1	9	greater flexibility in task assignment	Close control over software development processes and procedures
project monitoring and tracking	7	3	4	2	3	5	3	1,9	a lack of project monitoring and tracking	Project monitoring and tracking
micro project planning	3	5	6	5	6	5	6	1,9	macro project planning	Well-defined project charter and project plan that project stakeholders can understand
project managers motivate members using innovative ways	3	3	2	1	2	5	2	9	project managers do not act upon project members' motivation needs (template PM styles)	Motivating and managing performance
strong commitment from customers	2	2	4	1	6	1	1	9	a lack of commitment (lazy customers)	User involvement and participation
escalation management	1	3	4	1	4	3	2	9	a lack of escalation management	Effective escalation management process
culture diversity management	3	3	3	1,9	2	1	1	9,1	a homogeneous team composition	Team and culture building, Workforce planning

Participant #13

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
sensitive to culture and team dynamics issues	3	2	1	1	3	4	7	9	ignore culture and team dynamics issues	Team and culture building

extensive communication with executives & good governance (frequent report/update, visibility, and transparency)	1	6	3	2	6	8	2	9	infrequent & limited communication with executives (or only single executive) & poor governance (low visibility)	Management of sponsors and champions , Project transparency
shared project goals (or align project goals between project members)	2	7	3	1	3	7	1	9	a lack of project clarity	Well-defined project charter and project plan that project stakeholders can understand
high level of user engagement and participation (inclusive governance - the right to meaningfully participate in the decision-making processes)	1	3	3	1	2	7	8	9	low level of user engagement and participation	User involvement and participation
structural approach -well-defined and rigor standards, procedures, and structures	1	7	3	2	1	7	3	6	undefined and insufficient standards, procedures, and structures	Close control over software development processes and procedures, Standards for design

Participant #14

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
Develop or customize development frameworks and code generation tools to ensure	1	7	9	1	9	9	9	5	Leverage existing framework for development (e.g., J2EE, Spring)	Continuous attention to technical excellence, Leverage industry standards or best practices for IS projects, Tool/Software support

development efficiency and quality										
Poor communication between outsourcing partners and developers cause unsatisfied deliverables	N/A	7	N/A	9	1	N/A	N/A	1	Effective communication between outsourcing partners and developers	Close collaboration and communication between project members during development
documentation for compliance, which drains away engineering time (customers request detailed documentation)	7	3	8	9	2	7	7	1	Concise documentation for clearly capturing requirements; Sufficient documentation for design and deployment (if customers need comprehensive documentation, the work could be done after the implementation)	Careful and comprehensive documentation across all phases of development , Promote simplicity
adopt daily building tools to ensure software quality and accelerate development	9	9	9	1	3	9	9	9	a lack of daily building tools	Continuous integration, Tool/Software support

strive to impose standardized development processes	9	9	9	9	3	9	9	1	show no interests in standardized development processes	Close collaboration and communication between project members during development, Leverage industry standards or best practices for IS projects
Capable project managers and system analysts for managing scope and requirements	2	8	2	1	7	3	2	9	a team lacks capabilities to manage scope and requirements	Substantial attention to requirements analysis , Workforce planning

Participant #15

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
leverage external expertise to speed up the process	1	1	4	8	8	9	9	1	use internal expertise for extended functionalities (internal experts have better contextual knowledge)	Workforce planning, Project knowledge management
use proven concepts and practices for development, including business and technical solutions (following consultants' solutions)	1	9	4	9	5	4	5	1	formulate innovative solutions that fit specific user needs and requirements	Continuous attention to customer problems and satisfaction, Innovative design, Leverage industry standards or best practices for IS projects

unilateral development of requirement and design specification	1	6	4	8	9	9	9	2	collaborative approach to project completion	Close collaboration and communication between project members during development
centralized software architecture that guide IS projects	9	2	9	1	9	9	9	9	A lack of centralized architecture. Each module is developed individually.	Identify IT/business strategy and align projects with IT/business strategy

Participant #16

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
Well-developed project plans (e.g., clear goals and scope, assign right people in the right place)	4	4	9	1	1	3	5	9	vague project plans	Well-defined project charter and project plan that project stakeholders can understand, Workforce planning
involvement of some domain experts in a project	4	4	8	N/A	2	3	4	N/A	a team that is largely composed of inexperienced project members	Workforce planning
top management support , including explicit verbal statement.	3	3	8	1	1	3	5	9	implicit top management support	Management of sponsors and champions
positive team (reasonable expectation and adequate resources)	3	3	7	2	2	2	6	9	negative team (unreasonable expectation and inadequate resources)	Motivating and managing performance, Team and culture building
informal and flexible development process	4	4	7	3	3	3	6	9	rigid and formal development process	Close control over software development processes and procedures

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
communicate upfront and clearly among project members, champions, users, and other stakeholders.	5	5	7	2	2	2	5	9	communicate after the fact	Close collaboration and communication between project members during development, Management of sponsors and champions
inclusive collaboration (e.g., decisions made by a team)	4	4	7	2	2	2	6	9	exclusive collaboration (e.g., decisions are made by one or few experts)	Close collaboration and communication between project members during development, Team and culture building
defined processes for managing scope and requirements	4	4	6	3	3	3	6	9	scope creep (a lack of expectation management)	Close control over software development processes and procedures, Management of sponsors and champions
Clearly defined business success criteria -Projects work toward delivering business values use. -Metrics are used to measure and manage a project.	5	5	8	2	2	2	5	9	a lack of clearly defined business goals/success	Well-defined project charter and project plan that project stakeholders can understand, Identify IT/business strategy and align projects with IT/business strategy, Management and control via metrics
take whole business processes into consideration, which allows redefining the business processes	2	2	4	3	3	3	4	9	A sole focus on technology (an information system is built without considering business objectives)	Identify IT/business strategy and align projects with IT/business strategy

Participant #17

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization

strong vendor support	2	3	4	1	5	3	N/A	9	weak vendor support	Close collaboration and communication between project members during development
matured team capabilities (expertise in development and quality assurance, particularly)	4	3	3	1	2	2	3	9	inexperienced project team	Workforce planning
a lack of experienced business analysts, which causes issues regarding unclear requirements.	4	7	7	9	8	8	6	1	accurate, comprehensive, and up-to-date requirement definition (provided by experienced business analysts)	Continuous attention to customer problems and satisfaction, Workforce planning
effective communication among vendor, teammates, and all stakeholders	4	2	2	1	3	3	4	9	a lack of communication	Close collaboration and communication between project members during development, Management of sponsors and champions
iterative and incremental development with strict development procedures (e.g., Rational Unified Process)	3	2	2	1	2	4	7	7	Agile development (a lack of structured procedures for managing projects. Communication is essential for responding changes)	Close control over software development processes and procedures, Close collaboration and communication between project members during development, Iterative and incremental development

executable test plan (test cases, test tools, test scope, test environments)	2	3	3	1	2	4	7	9	incomplete and imprecise test plan	Defect prevention

Participant #18

Similar (1)	P1	P2	P3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
Pair programming -increases development efficiency -enhances product quality -facilitates communication	3	9	8	7	5	9	9	1	individual programmer -programmers may not prefer working in a pair (personality issues)	Close collaboration and communication between project members during development, Regular inspection and review of deliverables, Defect prevention, Team and culture building
Resolve technical issues by seeking support from external expertise	5	8	6	5	6	5	8	N/A	Resolve technical challenges by adopting established libraries, components, and modules (e.g., open sources code)	Leverage industry standards or best practices for IS projects, Reuse of code and components
project managers who have technical background can facilitate software development	1	9	9	3	3	6	3	9	PM without any technical background	Workforce planning
use industry standards or platforms to ensure scalability (e.g., J2EE)	9	1	1	1	1	9	9	9	developing an architecture from scratch	Leverage industry standards or best practices for IS projects
modular design	4	2	2	1	1	9	7	9	a lack of modular design	Modular design

source code control, including tools (e.g., Microsoft SourceSafe), code review, and unit testing before submitting code.	1	3	1	1	2	9	7	9	a lack of source code control (programmers keep code)	Close control over software development processes and procedures, Defect detection, Tool/Software support
automated build	9	1	7	1	1	9	7	9	manual build	Continuous integration, Tool/Software support
Project managers effectively communicate project issues with top management and customers and tries to address their issues.	7	1	1	3	6	4	4	9	Project managers rarely communicate with stakeholders. When issues come up, they often ignore project constraints and leaves pressure to project members	Management of sponsors and champions , Workforce planning

Participant #19

Similar (1)	p1	p2	p3	IDEAL	P4	P5	P6	SUB	Different (9)	Categorization
insufficient investigation on existing business practices	2	2	7	9	8	7	9	1	an emphasis on business analysis	Substantial attention to requirements analysis
dispersed collaboration	5	7	7	9	9	2	8	1	opportunities to work more closely and regularly	Close collaboration and communication between project members during development

Project managers motivate project members by well-defined expectation and transparent project status	6	6	3	1	2	2	2	9	project managers focus on timelines and do not provide appropriate support for project members and do not manage ISD processes and teams.	Motivating and managing performance, Project transparency, Team and culture building
accurate project estimate based on previous tasks	6	9	2	1	2	4	1	9	a lack of investigation into what involves in project	A sufficient level of requirements analysis , Project size, cost, and schedule estimation
project managers who are well versed in organizational constraints, procedures, and technical knowledge	9	9	3	1	2	4	1	9	Project managers who are lack of appropriate technical knowledge and knowledge of organizational policies	Plan the workforce
manage expectation of end-users and clients	9	9	2	1	2	7	2	9	a lack of communication and collaboration with users and clients (more old-fashioned waterfall way)	User involvement and participation, Management of sponsors and champions
standardized technologies, procedures, and coding practices	2	3	2	1	2	9	1	9	a lack of standards, including technologies and development processes	Close control over software development processes and procedures, Standards for design

Appendix E

Coding of Raw Concepts

ISD concepts	Interviewees' raw concepts[32]
Continuous attention to customer problems and satisfaction	▪ use proven concepts and practices for development, including business and technical solutions (or following consultants' solutions) vs. formulate innovative solutions that fit specific user needs and requirements (p#15)
Careful and comprehensive documentation across all phases of development	▪ careful design for the overall system architecture (with detailed technical documents) vs. experimental and ad-hoc design (refactoring is employed) (p#2) ▪ a lack of documentation vs. adequate update on documentation (with the assistance of automatic document generation) (p#2) ▪ informal specifications, such as emails and chat vs. well-defined specification (with too much details) (p#4) ▪ comprehensive requirement documentation at the design phase vs. comprehensive documentation at the implementation phase (p#9) ▪ requirements and design specification are comprehensively and accurately documented vs. requirements and design decisions are communicated and exchanged through informal meeting (p#10) ▪ documentation for compliance, which drains away engineering time (customers request detailed documentation) vs. concise documentation for clearly capturing requirements; Sufficient documentation for design and deployment (if customers need comprehensive documentation, the work could be done after the implementation) (p#14)
Close collaboration and communication between project members during development	▪ strong vendor support vs. strong community support (e.g., open source community) (p#1) ▪ close collaboration (e.g., open floor, stand-up meetings) vs. dispersed collaboration (p#3) ▪ task estimation based on team decisions vs. task estimation based on individual decisions (p#4) ▪ team-oriented decision-making vs. dictatorial decision-making (p#4) ▪ collaborative decision-making (consensus-based) vs. dictatorial decision-making (p#5) ▪ pair programming vs. linear development (p#8) ▪ share common languages/understanding between management and developers (or PM can act as a liaison) vs. a lack of communication between management and developers (p#11) ▪ Poor communication between outsourcing partners and developers cause unsatisfied deliverables vs. Effective communication between

[32] The text in the parenthesis is a participant ID.

	outsourcing partners and developers (p#14) ▪ unilateral development of requirement and design specification vs. collaborative approach to project completion (p#15) ▪ communicate upfront and clearly among project members, champions, users, and other stakeholders. vs. communicate after the fact (p#16) ▪ inclusive collaboration (e.g., decisions made by a team) vs. exclusive collaboration (e.g., decisions are made by one or few experts) (p#16) ▪ strong vendor support vs. weak vendor support (p#17) ▪ effective communication among vendor, teammates, and all stakeholders vs. a lack of communication (p#17) ▪ iterative and incremental development with strict development procedures (e.g., Rational Unified Process) vs. Agile development (a lack of structured procedures for managing projects. Communication is essential for responding changes) (p#17) ▪ pair programming (benefits: increases development efficiency, enhances product quality, and facilitates communication) vs. individual programmer ▪ programmers may not prefer working in a pair (personality issues) (p#18) ▪ dispersed collaboration vs. opportunities to work more closely and regularly (p#19)
Close control over software development processes and procedures	▪ less disciplined project management (a lack of project monitoring) vs. formal project management (strict monitoring and project control and risk assessment) (p#1) ▪ a lack of version control and source code management vs. version control (p#2) ▪ version/source code control vs. a lack of version/source code control (p#3) ▪ a lack of quality assurance management (e.g., developers conducted testing on their own) vs. systematic quality assurance management (e.g., QA role, QA plan, QA procedures) (p#6) ▪ single thread version control (CVS) vs. multithread version control (Mercurial) (management of multiple versions in parallel) (p#7) ▪ strictly follow the procedures and rules vs. flexible procedures and rules (p#11) ▪ planned task assignment (jobs should be done by specific person and time according to the plan) vs. greater flexibility in task assignment (p#12) ▪ structural approach -well-defined and rigor standards, procedures, and structures vs. undefined and insufficient standards, procedures, and structures (p#13) ▪ strive to impose standardized development processes vs. show no interests in standardized development processes (p#14) ▪ informal and flexible development process vs. rigid and formal development process (p#16) ▪ defined processes for managing scope and requirements vs. scope creep

	(a lack of expectation management) (p#16) ▪ iterative and incremental development with strict development procedures (e.g., Rational Unified Process) vs. Agile development (a lack of structured procedures for managing projects. Communication is essential for responding changes) (p#17) ▪ source code control, including tools (e.g., Microsoft SourceSafe), code review, and unit testing before submitting code. vs. a lack of source code control (programmers keep code) (p#18) ▪ standardized technologies, procedures, and coding practices vs. a lack of standards, including technologies and development processes (p#19)
Collective ownership for development processes and outcomes	▪ collective ownership of project, such as visible status, report, and outcomes vs. a lack of knowledge(awareness) of overall status and outcomes (p#5)
Conscious efforts to make project size, cost, and schedule estimation	▪ task estimation based on team decisions vs. task estimation based on individual decisions (p#4) ▪ task estimation based on story points vs. task estimation based on time (p#4) ▪ Software effort estimation based on few key players vs. Software effort estimation based on negotiation and consensus (p#8) ▪ accurate work breakdown structure vs. a lack of accurate project planning and estimation (p#10) ▪ appropriate buffer time estimation vs. a lack of buffer time estimation (p#11) ▪ component-based estimation vs. function points analysis (p#12) ▪ accurate project estimate based on previous tasks vs. a lack of investigation into what involves in project (p#19)
Continuous attention to technical excellence	▪ careful design for the overall system architecture (with detailed technical documents) vs. experimental and ad-hoc design (refactoring is employed) (p#2) ▪ test-driven development vs. non test-driven development (p#4) ▪ pair programming vs. linear development (p#8) ▪ code review vs. a lack of code review (p#8) ▪ involvement of technical lead/experts vs. a lack of attention to technical excellence (p#11) ▪ Develop or customize development frameworks and code generation tools to ensure development efficiency and quality vs. Leverage existing framework for development (e.g., J2EE, Spring) (p#14)
Continuous integration	▪ a lack of continuous integration vs. continuous integration (p#4) ▪ manual integration vs. automatic integration (p#4) ▪ infrequent release to the testing environment vs. frequent release to the testing environment (p#4) ▪ automated testing vs. a lack of automated testing (manual testing) (p#5) ▪ adopt daily building tools to ensure software quality and accelerate development vs. a lack of daily building tools (p#14)

	▪ automated build vs. manual build (p#18)
Defect detection	▪ rigorous quality control through testing vs. a lack of quality control (p#3) ▪ manual unit testing vs. automatic unit testing (p#4) ▪ infrequent release to the testing environment vs. frequent release to the testing environment (p#4) ▪ automated testing vs. a lack of automated testing (manual testing) (p#5) ▪ automated testing vs. manual testing (p#6) ▪ Extremely detailed and full-scale testing vs. high-level testing: functioning software is a primary goal (p#9) ▪ source code control, including tools (e.g., Microsoft SourceSafe), code review, and unit testing before submitting code. vs. a lack of source code control (programmers keep code) (p#18)
Defect prevention	▪ test-driven development vs. non test-driven development (p#4) ▪ pair programming for better software quality vs. individual programming (p#4) ▪ a lack of quality assurance management (e.g., developers conducted testing on their own) vs. systematic quality assurance management (e.g., QA role, QA plan, QA procedures) (p#6) ▪ pair programming vs. linear development (p#8) ▪ code review vs. a lack of code review (p#8) ▪ executable test plan (test cases, test tools, test scope, test environments) vs. incomplete and imprecise test plan (p#17) ▪ pair programming (benefits: increases development efficiency, enhances product quality, and facilitates communication) vs. individual programmer (p#18)
Develop capabilities of ISD professionals	▪ project rotation vs. no rotation (p#4) ▪ personal skill development vs. personal skill development isn't a focus (p#6)
Disciplined change evaluation and management	▪ new request management vs. a lack of request management (p#6)
Effective escalation management process	▪ resolve escalated issues vs. unclear escalating process (p#3) ▪ escalation management vs. a lack of escalation management (p#12)
Empowerment	▪ team-oriented decision-making vs. dictatorial decision-making (p#4) ▪ collaborative decision-making (consensus-based) vs. dictatorial decision-making (p#5) ▪ hands-on everything project management (close control) vs. loose control (p#7) ▪ micro project management (dedicated project managers pay attention to details and use a hands-on approach) vs. macro project management (p#11)
End users' welfare is the	None

major concern of IS development	
Frequent releases to customers	▪ incremental release (quick release) vs. full feature release (p#3) ▪ infrequent release to the production environment vs. frequent release to the production environment (p#4) ▪ small releases for demonstration, including working skeleton, prototyping vs. big bang releases (until the end of the project) -higher risk (p#5) ▪ shorter development cycle for frequent feedback vs. longer development cycle (p#7)
Identify IT/business strategy and align projects with IT/business strategy	▪ overarching enterprise architecture that documents business/IT plan, strategy, and roadmap vs. a lack of overarching enterprise architecture (p#9) ▪ overarching IT governance process in order to align business/IT plan, strategy, and roadmap vs. a lack of IT governance process (p#9) ▪ clear business strategies and visions vs. a lack of business strategies, visions, and objectives (p#11) ▪ centralized software architecture that guide IS projects vs. A lack of centralized architecture. Each module is developed individually (p#15) ▪ clearly defined business success criteria and projects work toward delivering business values (metrics are used to measure and manage a project) vs. a lack of clearly defined business goals/success (p#16) ▪ take whole business processes into consideration, which allows redefining the business processes vs. a sole focus on technology (an information system is built without considering business objectives) (p#16)
Innovative design	▪ careful design for the overall system architecture (with detailed technical documents) vs. experimental and ad-hoc design (refactoring is employed) (p#2) ▪ idea generation from multiple experts vs. dedicated human resources (p#3) ▪ iterative prototyping for learning and innovation vs. heavy up-front design, plan-driven design (p#6) ▪ use proven concepts and practices for development, including business and technical solutions (following consultants' solutions) vs. formulate innovative solutions that fitspecific user needs and requirements (p#15)
Iterative and incremental development	▪ evolving development processes vs. upfront thorough system analysis and design and the freeze requirements (p#1) ▪ iterative development vs. waterfall, sequential development (p#2) ▪ careful design for the overall system architecture (with detailed technical documents) vs. experimental and ad-hoc design (refactoring is employed) (p#2) ▪ iterative analysis and design vs. structural analysis and design, such as rational unified process and OOAD (p#3) ▪ incremental release (quick release) vs. full feature release (p#3)

	▪ linear development (plan-driven) vs. iterative and adaptive development (p#5) ▪ small releases for demonstration, including working skeleton, prototyping vs. big bang releases (until the end of the project, generally with higher risk (p#5) ▪ iterative prototyping for learning and innovation vs. heavy up-front design, plan-driven design (p#6) ▪ shorter development cycle for frequent feedback vs. longer development cycle (p#7) ▪ project management following the structures vs. improvised project management (chaos management) (p#7) ▪ ensure design "perfected" in the early stage of the project (with placement of abundant resources) vs. agile design based on customer feedback (p#11) ▪ iterative and incremental development with strict development procedures (e.g., Rational Unified Process) vs. Agile development (a lack of structured procedures for managing projects. Communication is essential for responding changes) (p#17)
Leverage industry standards or best practices for IS projects	▪ adoption of industry standard (e.g., ITIL, PMBOK, COBIT…) vs. no project-based standard (p#9) ▪ Develop or customize development frameworks and code generation tools to ensure development efficiency and quality vs. Leverage existing framework for development (e.g., J2EE, Spring) (p#14) ▪ strive to impose standardized development processes vs. show no interests in standardized development processes (p#14) ▪ use proven concepts and practices for development, including business and technical solutions (following consultants' solutions) vs. formulate innovative solutions that fit specific user needs and requirements (p#15) ▪ resolve technical issues by seeking support from external expertise vs. resolve technical challenges by adopting established libraries, components, and modules (e.g., open sources code) (p#18) ▪ use industry standards or platforms to ensure scalability (e.g., J2EE) vs. developing an architecture from scratch (p#18)
Management and control via metrics	▪ clearly defined business success criteria and projects work toward delivering business values (metrics are used to measure and manage a project) vs. a lack of clearly defined business goals/success (p#16)
Management of sponsors and champions	▪ traditional system analysis and design that focuses on system requirements vs. stakeholder analysis to determine the stakes and expectations of the stakeholders and communicate/negotiate better with the stakeholders (p#1) ▪ Get buy-in using business cases vs. no buy-in from stakeholders (p#1) ▪ responsive resource allocation vs. a lack of responsive resource allocation (p#8) ▪ project managers employ coordination and communication skills to resolve project issues that come from stakeholders vs. project managers do not have appropriate capability to deal with project issues (project

	managers receive issues from stakeholders and pass them to subordinates. They do not proactively solve problems) (p#8) ▪ minimum business commitment vs. maximum business commitment (p#9) ▪ the scope of the product is clearly defined and communicated (customers know the impact and cost of changes) vs. scope ambiguity causes problems in cost overrun and human resources management (p#10) ▪ share common languages/understanding between management and developers (or PM can act as a liaison) vs. a lack of communication between management and developers (p#11) ▪ extensive communication with executives & good governance (frequent report/update, visibility, and transparency) vs. infrequent & limited communication with executives (or only single executive) & poor governance (low visibility) (p#13) ▪ top management support , including explicit verbal statement. vs. implicit top management support (p#16) ▪ communicate upfront and clearly among project members, champions, users, and other stakeholders. vs. communicate after the fact (p#16) ▪ defined processes for managing scope and requirements vs. scope creep (a lack of expectation management) (p#16) ▪ effective communication among vendor, teammates, and all stakeholders vs. a lack of communication (p#17) ▪ project managers effectively communicate project issues with top management and customers and tries to address their issues. vs. project managers rarely communicate with stakeholders. When issues come up, they often ignore project constraints and leaves pressure to project members (p#18) ▪ manage expectation of end-users and clients vs. a lack of communication and collaboration with users and clients (more old-fashioned waterfall way) (p#19)
Modular design	▪ modular design vs. sequential or procedural design (tight module coupling) (p#2) ▪ Modular design vs. integrated (or coupled) design (p#6) ▪ limited reusability vs. high reusability design (component-based design) (p#8) ▪ flexible architecture vs. rigid architecture (p#10) ▪ modular design vs. a lack of modular design (p#18)
Motivating and managing performance	▪ individual-based reward vs. team-based reward (p#11) ▪ project managers motivate members using innovative ways vs. project managers do not act upon project members' motivation needs (template PM styles) (p#12) ▪ positive team (reasonable expectation and adequate resources) vs. negative team (unreasonable expectation and inadequate resources) (p#16) ▪ project managers motivate project members by well-defined expectation and transparent project status vs. project managers focus on timelines

	and do not provide appropriate support for project members and do not manage ISD processes and teams. (p#19)
Plan-driven development	▪ evolving development processes vs. Upfront thorough system analysis and design and the freeze requirements (p#1) ▪ iterative development vs. waterfall, sequential development (p#2) ▪ ▪ careful design for the overall system architecture (with detailed technical documents) vs. experimental and ad-hoc design (refactoring is employed) (p#2) ▪ iterative analysis and design vs. structural analysis and design, such as rational unified process and OOAD (p#3) ▪ linear development (plan-driven) vs. iterative and adaptive development (p#5) ▪ iterative prototyping for learning and innovation vs. heavy up-front design, plan-driven design (p#6) ▪ collaborative design with on-site customers (frequent communication with users) vs. design based on restricted requirement (early freezed documentation guide development) (p#7) ▪ project management following the structures vs. improvised project management (chaos management) (p#7) ▪ ensure design "perfect" in the early stage of the project (with placement of abundant resources) vs. agile design based on customer feedback (p#11) ▪ appropriate buffer time estimation vs. a lack of buffer time estimation (p#11)
Project knowledge management	▪ project and domain knowledge repository and transfer mechanisms, such as document repository, show and tell, and learning sessions vs. a lack of knowledge repository and transfer (p#6) ▪ leverage external expertise to speed up the process vs. use internal expertise for extended functionalities (internal experts have better contextual knowledge) (p#15)
Project monitoring and tracking	▪ less disciplined project management (a lack of project monitoring) vs. formal project management (strict monitoring and project control and risk assessment) (p#1) ▪ rigorous progress tracking -time tracking systems -visible progress chart vs. progress invisible to team members (only project managers are aware of progress) (p#6) ▪ frequent project review(e.g., status report) vs. Infrequent project review (p#8) ▪ systematic project monitor and review vs. causal project monitor and review (p#11) ▪ project monitoring and tracking vs. a lack of project monitoring and tracking (p#12)
Project transparency	▪ collective ownership of project, such as visible status, report, and outcomes vs. a lack of knowledge(awareness) of overall status and outcomes (p#5)

	▪ rigorous progress tracking-time tracking systems-visible progress chart vs. progress invisible to team members (only project managers are aware of progress) (p#6) ▪ extensive communication with executives & good governance (frequent report/update, visibility, and transparency) vs. infrequent & limited communication with executives (or only single executive) & poor governance (low visibility) (p#13) ▪ project managers motivate project members by well-defined expectation and transparent project status vs. project managers focus on timelines and do not provide appropriate support for project members and do not manage ISD processes and teams. (p#19)
Promote simplicity	▪ documentation for approval(mandatory and may not be useful) vs. documentation based on needs (useful) (p#5) ▪ extremely detailed and full-scale testing vs. high-level testing: functioning software is a primary goal (p#9) ▪ documentation for compliance, which drains away engineering time (customers request detailed documentation) vs. Concise documentation for clearly capturing requirements; Sufficient documentation for design and deployment (if customers need comprehensive documentation, the work could be done after the implementation) (p#14)
Reflect on improvement at regular intervals	None
Regular inspection and review of deliverables	▪ code review vs. a lack of code review (p#8) ▪ pair programming (benefits: increases development efficiency, enhances product quality, and facilitates communication) vs. individual programmer (p#18)
Reuse of code and components	▪ limited reusability vs. high reusability design (component-based design) (p#8) ▪ resolve technical issues by seeking support from external expertise vs. Resolve technical challenges by adopting established libraries, components, and modules (e.g., open sources code) (p#18)
Standards for design	▪ a lack of control for programming vs. setting programming standard and establishing norms of good programming (p#2) ▪ vendor-owned development & production environment vs. client-owned development & product environment (p#3) ▪ cloud-based development environment vs. local (internal) network development environment (p#3) ▪ coding standards vs. a lack of coding standards (p#10) ▪ structural approach-well-defined and rigor standards, procedures, and structures vs. undefined and insufficient standards, procedures, and structures (p#13) ▪ standardized technologies, procedures, and coding practices vs. a lack of standards, including technologies and development processes (p#19)
Substantial attention to	▪ Capable project managers and system analysts for managing scope and

requirements analysis	requirements vs. a team lacks capabilities to manage scope and requirements (p#14) ▪ a lack of experienced business analysts, which causes issues regarding unclear requirements. vs. accurate, comprehensive, and up-to-date requirement definition (provided by experienced business analysts) (p#17) ▪ insufficient investigation of existing business practices vs. an emphasis on business analysis (p#19) ▪ accurate project estimate based on previous tasks vs. a lack of investigation into what involves in project (p#19)
Team and culture building	▪ informal work relationships between project members. Project members know each other well vs. formal work relationships between project members (minimal social interaction and a lack of understanding) (p#10) ▪ individual-based reward vs. team-based reward (p#11) ▪ supportive leadership style vs. subordinates are criticized and reprimanded when they do something incorrectly (p#11) ▪ culture diversity management vs. a homogeneous team composition (p#12) ▪ sensitive to culture and team dynamics issues vs. ignore culture and team dynamics issues (p#13) ▪ positive team (reasonable expectation and adequate resources) vs. negative team (unreasonable expectation and inadequate resources) (p#16) ▪ inclusive collaboration (e.g., decisions made by a team) vs. exclusive collaboration (e.g., decisions are made by one or few experts) (p#16) ▪ pair programming (benefits: increases development efficiency, enhances product quality, and facilitates communication) vs. individual programmer (p#18) ▪ project managers motivate project members by well-defined expectation and transparent project status vs. project managers focus on timelines and do not provide appropriate support for project members and do not manage ISD processes and teams. (p#19)
Explicit recognition and management of risk	▪ less disciplined project management (a lack of project monitoring) vs. formal project management (strict monitoring and project control and risk assessment) (p#1)
Tool/Software support	▪ a lack of documentation vs. adequate update on documentation (with the assistance of automatic document generation) (p#2) ▪ a lack of version control and source code management vs. version control (p#2) ▪ version/source code control vs. a lack of version/source code control

	(p#3) ▪ manual unit testing vs. automatic unit testing (p#4) ▪ manual integration vs. automatic integration (p#4) ▪ automated testing vs. a lack of automated testing (manual testing) (p#5) ▪ automated testing vs. manual testing (p#6) ▪ single thread version control (CVS) vs. multithread version control (Mercurial) (management of multiple versions in parallel) (p#7) ▪ develop or customize development frameworks and code generation tools to ensure development efficiency and quality vs. leverage existing framework for development (e.g., J2EE, Spring) (p#14) ▪ adopt daily building tools to ensure software quality and accelerate development vs. a lack of daily building tools (p#14) ▪ source code control, including tools (e.g., Microsoft SourceSafe), code review, and unit testing before submitting code. vs. a lack of source code control (programmers keep code) (p#18) ▪ automated build vs. manual build (p#18)
User involvement and participation	▪ intensive user involvement -users work closely with developers and help testing -users involve throughout the whole project cycle vs. less user involvement -users only participate in the inititation and closing stage. (p#1) ▪ infrequent feedback from product owners vs. frequent feedback from product owners (p#4) ▪ end-user collaboration (with intensive involvement) vs. a lack of end-user collaboration (speculation of requirements and issues from developer teams) (p#5) ▪ on-site customers vs. No access to users -rely on documentation and specification (p#6) ▪ collaborative design with on-site customers (frequent communication with users) vs. design based on restricted requirement (early freezed documentation guide development) (p#7) ▪ open and transparent communication with customers on IT solutions vs. no communication with customers on IT solutions (p#9) ▪ effective and open communication with customers (the goal is to obtain accurate requirements) vs. poor communication between customers and project teams. This causes misunderstanding of requirements and inaccurate expectation of deliverables (p#10) ▪ ensure design "perfect" in the early stage of the project (with placement of abundant resources) vs. agile design based on customer feedback (p#11) ▪ strong commitment from customers vs. a lack of commitment (lazy customers) ▪ high level of user engagement and participation (inclusive governance - the right to meaningfully participate in the decision-making processes) vs. low level of user engagement and participation (p#13) ▪ manage expectation of end-users and clients vs. a lack of communication and collaboration with users and clients (more old-fashioned waterfall

	way) (p#19)
Well-defined project charter and project plan that project stakeholders can understand	▪ The scope of the product is clearly defined and communicated. Customers know the impact and cost of changes vs. Scope ambiguity causes problems in cost overrun and human resources management (p#10) ▪ micro project planning vs. macro project planning (p#12) ▪ shared project goals (or align project goals between project members) vs. a lack of project clarity (p#13) ▪ Well-developed project plans(e.g., clear goals and scope, assign right people in the right place) vs. vague project plans (p#16) ▪ clearly defined business success criteria and projects work toward delivering business values (metrics are used to measure and manage a project) vs. a lack of clearly defined business goals/success (p#16)
Workforce planning	▪ leverage external expertise for complicated technical issues vs. Solve complicated technical issues by internal knowledge (p#1) ▪ a lack of competent project members vs. good team composition (involving senior developers and looking for competent developers) (p#2) ▪ strong knowledge/expert support vs. absence of support (p#3) ▪ scalable human resource management vs. dedicated human resources (p#3) ▪ idea generation from multiple experts vs. dedicated human resources (p#3) ▪ project rotation vs. no rotation (p#4) ▪ a lack of quality assurance management (e.g., developers conducted testing on their own) vs. systematic quality assurance management (e.g., QA role, QA plan, QA procedures) (p#6) ▪ QAs have sophisticated domain and application knowledge vs. QAs have testing knowledge and techniques alone (p#8) ▪ project managers employ coordination and communication skills to resolve project issues that come from stakeholders vs. project managers do not have appropriate capability to deal with project issues (project managers receive issues from stakeholders and pass them to subordinates. They do not proactively solve problems) (p#8) ▪ dedicated human resources for each project vs. human resource pool (p#11) ▪ a team is composed of experienced developers vs. a team is largely composed of inexperienced developers (p#11) ▪ independent testers vs. developers as testers (p#11) ▪ culture diversity management vs. a homogeneous team composition (p#12) ▪ capable project managers and system analysts for managing scope and requirements vs. a team lacks capabilities to manage scope and requirements (p#14) ▪ leverage external expertise to speed up the process vs. use internal expertise for extended functionalities (internal experts have better

	contextual knowledge) (p#15) ▪ well-developed project plans (e.g., clear goals and scope, assign right people in the right place) vs. vague project plans (p#16) ▪ involvement of some domain experts in a project vs. a team that is largely composed of inexperienced project members (p#16) ▪ matured team capabilities (expertise in development and quality assurance, particularly) vs. inexperienced project team (p#17) ▪ a lack of experienced business analysts, which causes issues regarding unclear requirements. vs. accurate, comprehensive, and up-to-date requirement definition (provided by experienced business analysts) (p#17) ▪ project managers who have technical background can facilitate software development vs. PM without any technical background (p#18) ▪ Project managers effectively communicate project issues with top management and customers and tries to address their issues. vs. Project managers rarely communicate with stakeholders. When issues come up, they often ignore project constraints and leaves pressure to project members (p#18) ▪ project managers who are well versed in organizational constraints, procedures, and technical knowledge vs. Project managers who are lack of appropriate technical knowledge and knowledge of organizational policies (p#19)

Appendix F

Personal Background Information

(A) The following questions are designed to understand your background. Could you give us a few bits of information about yourself so that we can put your responses in greater context? Your answers to these questions are strictly confidential. For each of the following questions, please check or fill in the appropriate information.

1. Years of experience in the IS-relevant job: _______________
   (1) Years of experience in programming: _______
   (2) Years of experience in business/system analysis: _______
   (3) Years of experience in business/system architecture: _______
   (4) Years of experience in project management: _______
   (5) Years of experience in testing: _______
2. Please check the boxes that best describe your confidence in your expertise in the following ISD activities

	Not At all	Just a Little	Somewhat	Moderately	Quite a Bit	Considerably	Extremely
Project Management	☐	☐	☐	☐	☐	☐	☐
Business Analysis	☐	☐	☐	☐	☐	☐	☐
System Analysis	☐	☐	☐	☐	☐	☐	☐
Business Architecture	☐	☐	☐	☐	☐	☐	☐
System Architecture	☐	☐	☐	☐	☐	☐	☐
Programming	☐	☐	☐	☐	☐	☐	☐
Testing	☐	☐	☐	☐	☐	☐	☐
Other, please specify: _______	☐	☐	☐	☐	☐	☐	☐

3. Current Job Title: _______________________
4. Which range includes your age?
   ☐ 18-24 ☐ 25-34 ☐ 35-44 ☐ 45-54
   ☐ 55-64 ☐ 65 or older ☐ Prefer not to answer

5. Please indicate your gender: □ Male □ Female □ Prefer not to answer
6. What is the highest level of education that you have completed?
   - □ Secondary/High School
   - □ College/University, please indicate what program you took (e.g., computer science) __________
   - □ Post-graduate Studies, please indicate what program you took (e.g., computer science) __________

Appendix G

Project Background Information

(A) The following questions are designed to understand the project background. Your answers to these questions are strictly confidential. For each of the following questions, please check or fill in the appropriate information.

1. Please give the project a code that you will use in this study: _______________

2. Which of the following best describes the type of project?
   - ☐ In-house New Development
   - ☐ Packaged Software Implementation
   - ☐ Enhancement of Existing Software
   - ☐ Other _______________

3. Which of the following best describes your role in this project? (Check all that apply)
   ☐ Business/System Analyst ☐ Business/System Architect ☐ Project Manager
   ☐ Programmer ☐ Senior Business Management ☐ Senior IT Management
   ☐ Tester ☐ Users ☐ Consultant ☐ Other _______________

4. Which of the following categories best describes the industry your project is in?
   ☐ Consulting ☐ Education ☐Finance and Insurance ☐ Government
   ☐ Healthcare ☐ Manufacturing ☐ Retail ☐ Software ☐ Telecom/network
   ☐ Transportation ☐ Other _______________

5. How many project members are/were in this project team? _______________

6. What is the project budget (Canadian dollars)? _______________

7. What is the project duration (months)? _______________

Appendix H

Definitions of ISDP Concepts

CID	Concept	Definition
1	Explicit recognition and management of risk	Identifying, analyzing, and responding to project risk. Work practices include: scenario planning, SWOT analysis, throwaway prototyping, risk monitoring, actively managing top-10 risk list, etc.
2	Project monitoring and tracking	Track actual results and performances against the project plans in a timely manner. When deviations exist, corrective actions are taken. Work practices include: "earned value" project tracking, daily standup meetings, etc.
3	Team and culture building	Team and culture building improves coordination and interaction among team members and maximizes the capabilities of each team member. Practices include: teamwork exercises and fun occasions (e.g., pot luck lunches, sports events), rewarding teamwork, and recognition by leaders.
4	Promote simplicity	The identification and elimination of waste from the design and process. Waste can mean design that is difficult to change, extra features, waiting, task switching, extra processes, partially done work, defects, etc.
5	Management and control via metrics	Quantify and measure project progress (daily or less frequently), project members' performance, and product quality. Example measures include: code coverage, customer satisfaction, level of product testability, or team overtime hours. Teammates and stakeholders view measures, provide feedback, and suggest improvements.
6	Standards for design	Mandatory requirements are employed and enforced to create a disciplined uniform approach to software development. Practices include: unified graphical models, common coding standards, coding style guideline, etc.
7	Collective ownership for development processes and outcomes	Every team member contributes to and is responsible for the quality and improvement of a software system/ subsystem and for problems and flaws.
8	End users' welfare is the major concern of IS development	Analysis and design processes should ensure the end user's satisfaction and welfare. Different approaches, such as in-depth interview, job shadowing, and close observation of user groups, should be adopted to understand how an information system can affect work life and job satisfaction.

9	Defect detection	The process of finding and fixing defects after a product is built. Different test techniques, such as black-box testing and white-box testing, are applied to all levels of software testing: unit, integration, system, and acceptance. The concept of defect detection also implies that the flawed development process that generated those defects is left uncorrected.
10	Motivating and managing performance	A work environment that has adequate resources, reasonable compensation, and does not impede or distract from job performance should be provided to motivate project members. Further, individuals and team performance objectives should be established and evaluated in order to improve the performance. Practices include: reasonable compensation and reward mechanisms, sustainable working hours, praising project members, avoiding criticisms in open forums, and setting achievable goals.
11	User involvement and participation	Users provide feedback and help the development team but also actively engage in the whole IS development process. Examples include: on-site customers for close, daily cooperation and decision-making involvement, involvement of users in test development and validation, end-of-iteration customer focus groups, etc.
12	Defect prevention	The root causes of defects are identified and prevented from recurring. This involves collecting defect data in a defect repository, analyzing and identifying the root causes of the most severe defects, and applying a systematic methodology to prevent these defects from recurring. Practices include: test-driven development, automated testing, root-cause analysis, collection of statistics on defects, adoption of design patterns, and pair programming. Different from defect detection, defect prevention evaluates all defects and to identify, fix, and prevent other similar problems.
13	Continuous attention to technical excellence	Technical excellence ensures created products respond to customer needs, within established time and cost constraints. It also reduces the cost of change so the product remains responsive to future customer needs. Practices include: refactoring, design patterns, etc.
14	Well-defined project charter and project plan that project stakeholders can understand	A well-defined project charter describes project purpose, summary milestone schedule, summary budget, high-level project descriptions and requirements, high-level risks, project success criteria, project approval requirements, and involved stakeholders. A well-defined project plan guides project execution and control. It explains in detail how and when to fulfill the project objectives by showing the major products, milestones, activities, and resources required for the project. All stakeholders must fully understand the

		above information.
15	Project transparency	Problems, risks, and issues are shared openly and transparently. Different forms of status information are presented through a tracking and report system, Task Kanban board, the burn down chart, and the iteration status board.
16	Conscious efforts to make project size, cost, and schedule estimation	Determining how complex a project is, how much it will cost, and how long a project will take. Careful estimation involves the right people (i.e., experts) making estimation at right time (i.e., requirements are defined). Estimation techniques include: COCOMO2, function point analysis, planning poker technique with story points (group-based estimates with historical data), a time boxing technique, etc.
17	Identify IT/business strategy and align projects with IT/business strategy	Information systems selected for development should support an organization's IT/business strategies. Such elements provide a baseline to manage projects and ensure that a delivered project reflects business values.
18	Empowerment	Each team member is fully empowered to complete assigned tasks. Each member is expected to work for the benefit of the whole team.
19	Substantial attention to requirements analysis	Ensures that all business rules are captured, business processes are understood, and business needs are integrated into the application design.
20	Plan-driven development	Plan-driven development is based on careful upfront planning and design, formal contracts with customers, and following a plan. Also, plan-driven development is characterized by early scope freeze. If changes are needed or emergency situations occur, management reserves (i.e., certain amount of resources are dedicated to contingency) are used.
21	Regular inspection and review of deliverables	Regular inspection and review is used to detect errors, development standard violations, and other problems. Commonly inspected artifacts include: requirements documents, design, code, and test plans. Work practices include: independent audits (walkthroughs and inspections of design and code), formal procedure (e.g., review or handover with sign-off), sprint review, etc.
22	Close collaboration and communication between project members during development	All development activities should include frequent, fluid communication and integrated actions among internal project members. Practices include: pair programming, working in a common project room, regular meetings to solve problems, establishing a communication/coordination policy, etc.
23	Continuous attention to	To deliver value to customers, problems must be addressed and customer satisfaction achieved. These actions

	customer problems and satisfaction	strengthen business goals. Such attention also creates successful projects because customer expectations and requirements are met.
24	Effective escalation management process	Rules governing who is authorized to take charge when problems or change arise should be planned and executed, as well as how issues are flagged and escalated.
25	Close control over software development processes and procedures	A sequence of steps is defined to govern what people do and should be rigorously followed during software development. Work practices include: document approval by the appropriate authority after each phase, code submission procedures, procedures for user acceptance, etc.
26	Continuous integration	Members of a team integrate their work frequently using work practices such as daily build. Each integration is verified by an automated build that run all automated tests and detect integration errors as quickly as possible.
27	Management of sponsors and champions	Involves/Includes reporting, communications, and expectation management with sponsors, champions, and top management about project costs, efforts, and benefits. This ensures proper support and resources, i.e., sponsors and champions actively support the project and budget accordingly. Work practices include the use of communication plans, well-timed deliverables, and stakeholder assessment.
28	Careful and comprehensive documentation across all phases of development	The whole development life cycle of must be properly documented, including project plans, models, requirements, code, test cases, and user manuals so that project members and users can review them. Documentation must also be updated whenever necessary.
29	Leverage industry standards or best practices for IS projects	Adopt or tailor formalized industry standards to ensure software delivery in areas such as software platform (e.g., J2EE), enterprise architecture (e.g., COBIT, ITIL), software development (e.g., CMMI, Rational Unified Process), and project management (PMBOK, PRINCE2).
30	Iterative and incremental development	The overall software development life cycle is composed of several iterations. Each iteration is a self-contained mini-project composed of activities such as requirement analysis, design, programming, and testing. The release, which is made at the end of an iteration, should be a stable, integrated and tested partially-completed system (which may be only an internal release). The concept of growing a system via iterations has been called Iterative and Incremental Development. It assumes that changes are inevitable; therefore, teams should welcome and accommodate for future change by building flexibility (e.g. parameterizing, deferring critical design decisions until the last possible

		moment).
31	Reuse of code and components	Developed and/or purchased code, modules, and components are managed and reused. Reused elements are all thoroughly tested and provide thorough information about the interface.
32	Reflect on improvements at regular intervals	A team reflects upon successes and challenges in recent work and adjusts development practices accordingly. Work practices include retrospective meetings, continuous process improvement toward an SEI CMM maturity level, and Kaizen improvement.
33	Project knowledge management	Essential knowledge about customers, domains, process, designs, templates, and plans that have been used to create software and services are retained and shared among project members.
34	Modular design	Each module can be implemented separately, and a change to one module has minimal impact on other modules. A modular design reduces complexity and facilitates changes. The result is efficient and effective implementation that encourages parallel development of different parts of different system parts.
35	Workforce planning	A project team plans, manages, and evaluates the workforce activities required to meet its capability objectives for (their) projects. Practices include: selecting a project team, establishing basic staffing practices, developing plans for workforce development, setting and tracking competency objectives in the workforce, etc.
36	Tool/Software support	Tools and software are used to support development process and project management activities. Tools and software include: graphical facilities for modeling and design, data dictionary, automated documentation, code generation, critical path scheduling with resource availability, report generators, defect tracking systems, etc.
37	Frequent releases to customers	Frequent releases can result in faster market releases for new services. Additionally, frequently obtained feedback from customers helps develop an information system that better fits customer needs.
38	Develop capabilities of IS development professionals	Immediate training and/or mentoring needs are identified to ensure that individuals have the necessary skills to accomplish their work. Practices include creating a learning environment, encouraging individuals to establish their own professional development programs, etc.
39	Innovative design	Unlike traditional design that tends to quickly converge on a solution, innovative design considers multiple solutions at the onset, tests their feasibility, and gradually converges on a final solution. Work practices include rapid prototyping, set-based design approach, etc.

40	Disciplined change evaluation and management	When any change occurs (e.g. resources, scope, design, or cost), change control identifies the impact on an information system, cost, and schedule. It evaluates benefits and costs and identifies alternatives. Then, the change control board accepts or rejects changes. All stakeholders should be aware of the change management policy and agree to its conditions.

Appendix I

Frequency of the Concepts Mentioned in RGT Interviews

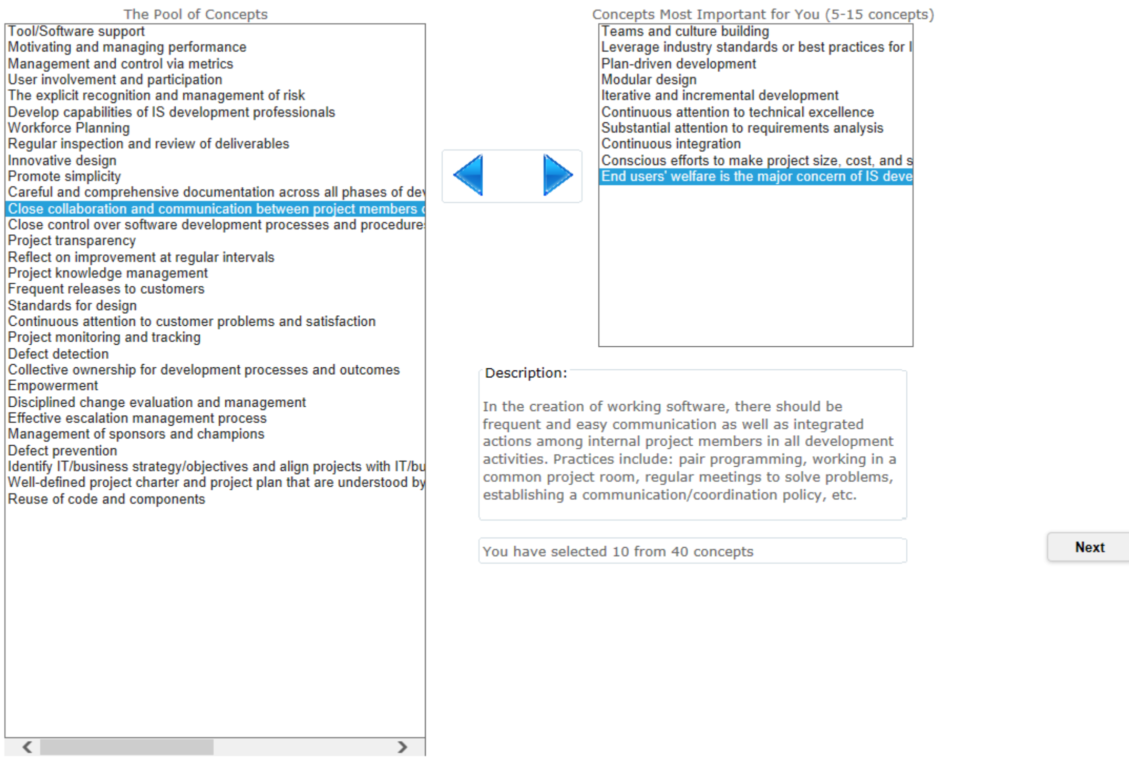
ISD concepts	Frequency	Percentage
C35. Workforce planning	22	9.48%
C22. Close collaboration and communication between project members during development	16	6.90%
C25. Close control over software development processes and procedures	15	6.47%
C27. Management of sponsors and champions	13	5.60%
C20. Plan-driven development	12	5.17%
C30. Iterative and incremental development	12	5.17%
C11. User involvement and participation	11	4.74%
C36. Tool/Software support	10	4.31%
C3. Team and culture building	9	3.88%
C16. Conscious efforts to make project size, cost, and schedule estimation	7	3.02%
C6. Standards for design	6	2.59%
C9. Defect detection	6	2.59%
C12. Defect prevention	6	2.59%
C23. Continuous attention to technical excellence	6	2.59%
C17. Identify IT/business strategy and align projects with IT/business strategy	6	2.59%
C26. Continuous integration	6	2.59%
C28. Careful and comprehensive documentation across all phases of development	6	2.59%
C29. Leverage industry standards or best practices for IS projects	6	2.59%
C2. Project monitoring and tracking	5	2.16%
C14. Well-defined project charter and project plan that project stakeholders can understand	5	2.16%
C34. Modular design	5	2.16%
C10. Motivating and managing performance	4	1.72%
C15. Project transparency	4	1.72%

C18. Empowerment	4	1.72%
C19. Substantial attention to requirements analysis	4	1.72%
C37. Frequent releases to customers	4	1.72%
C39. Innovative design	4	1.72%
C4. Promote simplicity	3	1.29%
C21. Regular inspection and review of deliverables	2	0.86%
C24. Effective escalation management process	2	0.86%
C31. Reuse of code and components	2	0.86%
C33. Project knowledge management	2	0.86%
C38. Develop capabilities of IS development professionals	2	0.86%
C1. Explicit recognition and management of risk	1	0.43%
C5. Management and control via metrics	1	0.43%
C7. Collective ownership for development processes and outcomes	1	0.43%
C13. Continuous attention to customer problems and satisfaction	1	0.43%
C40. Disciplined change evaluation and management	1	0.43%
C8. End users' welfare is the major concern of IS development	0	0.00%
C32. Reflect on improvements at regular intervals	0	0.00%

Appendix J

Measurement of ISDP Mental Models

(1) Please first look through the list of information system development (ISD) concepts in the left box (called the Pool of Concepts) and definitions on the lower right-hand corner of the screen. Then, select 10 concepts which you feel are of **most importance for successfully managing ISD projects** (press the right and left arrow to move concepts right or left). **Please ensure that you understand the meanings of the concepts correctly and pick the concepts that immediately come to mind.** There is no right or wrong answer. We just would like to know your opinions on ISD project management

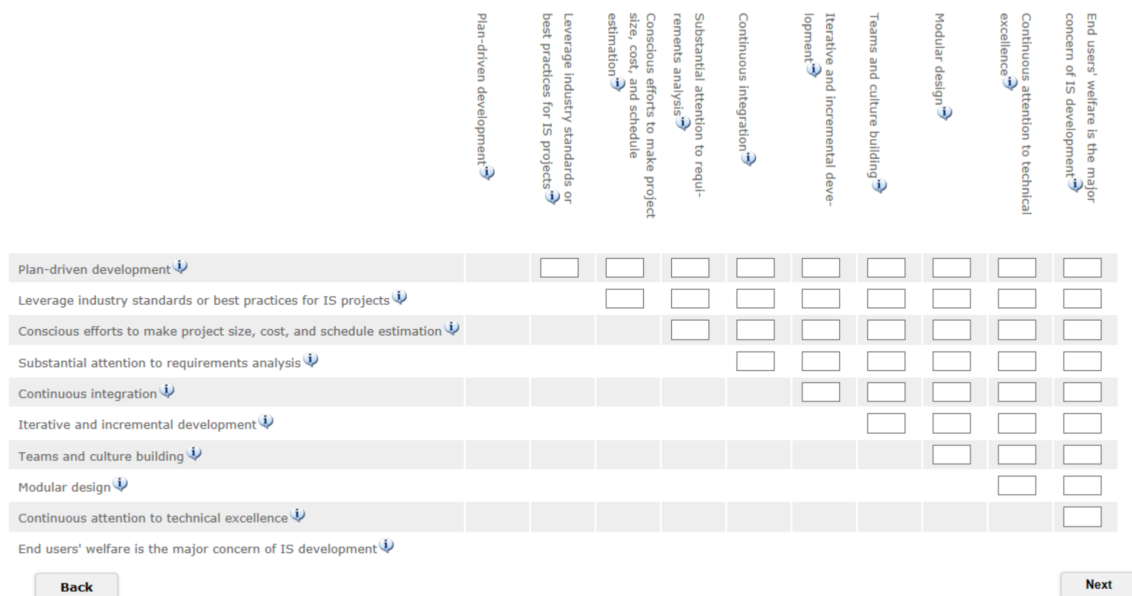


(2) Below are the concepts that you indicated are most important for successfully managing information systems development projects. These concepts are typically applied to the management of ISD projects for achieving certain project outcome(s). We would like to learn your opinions on how related these concepts are. There is no right or wrong answer. Based on your experience and knowledge, please rate how related each of these concepts is to each of the other concepts. Please use the following scale:

1	2	3	4	5	6	7	8	9
Unrelated								Highly Related

Example: In the upper left-hand cell (in the second column), you are asked to indicate how similar **concept** 1 (e.g., project transparency – the concept varies depending on your selection in the previous page) is to **concept** 2 (e.g., user involvement and participation) in achieving certain project outcome(s). You can enter a number from 1 to 9 in the cell.

Note: you can click on the information icon (ⓘ) to see a description of a concept. Also, when you move your mouse cursor over a cell, the associated concept names will appear.



	Plan-driven development	Leverage industry standards or best practices for IS projects	Conscious efforts to make project size, cost, and schedule estimation	Substantial attention to requirements analysis	Continuous integration	Iterative and incremental development	Teams and culture building	Modular design	Continuous attention to technical excellence	End users' welfare is the major concern of IS development
Plan-driven development		☐	☐	☐	☐	☐	☐	☐	☐	☐
Leverage industry standards or best practices for IS projects			☐	☐	☐	☐	☐	☐	☐	☐
Conscious efforts to make project size, cost, and schedule estimation				☐	☐	☐	☐	☐	☐	☐
Substantial attention to requirements analysis					☐	☐	☐	☐	☐	☐
Continuous integration						☐	☐	☐	☐	☐
Iterative and incremental development							☐	☐	☐	☐
Teams and culture building								☐	☐	☐
Modular design									☐	☐
Continuous attention to technical excellence										☐
End users' welfare is the major concern of IS development										

Back Next

Note. The order of questions will be randomized across respondents.

Appendix K

Frequency of the Concepts Selected by Survey Respondents

CID	Description	Frequency	Percentage
C11	User involvement and participation	40	42.11%
C6	Standards for design	38	40.00%
C9	Defect detection	35	36.84%
C2	Project monitoring and tracking	34	35.79%
C23	Continuous attention to customer problems and satisfaction	33	34.74%
C10	Motivating and managing performance	31	32.63%
C21	Regular inspection and review of deliverables	30	31.58%
C35	Workforce planning	29	30.53%
C33	Project knowledge management	28	29.47%
C3	Team and culture building	27	28.42%
C12	Defect prevention	27	28.42%
C22	Close collaboration and communication between project members during development	27	28.42%
C31	Reuse of code and components	27	28.42%
C39	Innovative design	27	28.42%
C26	Continuous integration	26	27.37%
C28	Careful and comprehensive documentation across all phases of development	26	27.37%
C34	Modular design	26	27.37%
C30	Iterative and incremental development	25	26.32%
C36	Tool/Software support	25	26.32%
C14	Well-defined project charter and project plan that project stakeholders can understand	24	25.26%
C17	Identify IT/business strategy and align projects with IT/business strategy	24	25.26%

C4	Promote simplicity	22	23.16%
C19	Substantial attention to requirements analysis	22	23.16%
C38	Develop capabilities of IS development professionals	22	23.16%
C18	Empowerment	21	22.11%
C15	Project transparency	20	21.05%
C20	Plan-driven development	19	20.00%
C25	Close control over software development processes and procedures	19	20.00%
C1	Explicit recognition and management of risk	18	18.95%
C8	End users' welfare is the major concern of IS development	18	18.95%
C37	Frequent releases to customers	18	18.95%
C24	Effective escalation management process	17	17.89%
C16	Conscious efforts to make project size, cost, and schedule estimation	16	16.84%
C7	Collective ownership for development processes and outcomes	15	15.79%
C13	Continuous attention to technical excellence	14	14.74%
C40	Disciplined change evaluation and management	14	14.74%
C5	Management and control via metrics	13	13.68%
C32	Reflect on improvements at regular intervals	13	13.68%
C29	Leverage industry standards or best practices for IS projects	12	12.63%
C27	Management of sponsors and champions	10	10.53%

Appendix L

Distance Ratio Formula and an Example

The general idea behind the distance ratio formula is calculated by summing up differences between the maps and divide the sum by the total number of possible differences. Differences between maps include the absence of a concept in one map, and different ratings for common concepts. We also incorporate missing values for common concepts as differences (see discussion below).

We simplified Markoczy and Goldberg's (1995), as demonstrated in Figure 21, by removing some parameters irrelevant to the study (e.g., the number of possible polarities) and revised the formula to include the condition where common concepts exist, but there is no relationship (i.e. missing values) (see clause (ii))[33]. In other words, two people may choose the same concepts, but only one may indicate a relationship between them. This is classified as a difference and is given a value of 1. If both have missing values, we do not know whether differences exist between the two maps, and no difference is added (i.e., 0). As for clause (iii), this circumstance occurs when one person selects a concept and another does not. Markoczy and Goldberg (1995) wish to avoid inflating the differences in this situation by attributing values to the relationship, but still consider it a difference. Therefore, 1 rather than m_{ij} is added. Clause (iv) is simply used to calculate differences between common concepts.

$$\text{DR (A, B)} = \frac{\sum_{i=1}^{p}\sum_{j=1}^{p}\left|a_{ij}^{*}-b_{ij}^{*}\right|}{\beta(p_c^2)+2p_c(p_{uA}+p_{uB})+(p_{uA}^2+p_{uB}^2)-(\beta(p_c)+p_{uA}+p_{uB})}$$

$$where\ m_{ij}^{*} = \begin{cases} o\ (i) if\ i=j \\ 1\ (ii)\ if\ either\ a_{ij} or\ b_{ij} has\ a\ missing\ value\ and\ i,j \in Pc \\ 1\ (iii) if\ m_{ij} \neq 0\ and\ i\ or\ j\ \notin\ Pc \\ m_{ij}\ (iv) otherwise \end{cases}$$

Where p is total number of possible concepts, $a_{ij} or\ b_{ij}$ is the value of the *i*th row and *j*th row of A or B, *Pc* is the set of concepts common to both maps, p_c is the number of such concepts. p_{uA} is the number of concepts unique to map A and p_{uB} is the number of concepts unique to B. m_{ij} is the value of the ith row jth column in the extended association

[33] The source code, downloaded from http://www.goldmark.org/jeff/programs/distrat/software, was revised.

matrix M. β is the maximum strength between concepts.

Figure 21 Revised Distance Ratio Formula

To help readers understand the formula, we have provided a sample example. Two fabricated maps are presented in Figure 2. For the sake of simplicity, only three concepts are represented. The total number of possible differences is equal to 8 *(2*2) + (2*2*(1+1))+(1+1)-(8*2+1+1) = 24 (i.e., possible number of differences between common concepts ((1,1), (1,2), (2,1), (2,2)) + possible number of differences between common and unique concepts (Map A: (1,3), (2,3), (3,2), (3,1); Map B: (1,4), (2,4), (4,2), (4 ,1)) + possible number of differences for unique concepts ((3,3) and (4,4) – the diagonal of the matrix (1,1), (2,2), (3,3), (4,4)). The actual difference is | (8-1) |+ | (1-8)|+8 = 22 (i.e., difference of (1,2) + difference of (2,1) + unique concepts (Map A: (1,3), (2,3), (3,2), (3,1); Map B: (1,4), (2,4), (4,2), (4 ,1)). Thus, the distance ratio between a project manager and project member is 22/24 = 0.9167.

Map A: Project Manager

Concept	C1	C2	C3
C1	0	1	5
C2	1	0	5
C3	5	5	0

Map B: Developer

Concept	C1	C2	C4
C1	0	8	5
C2	8	0	5
C4	5	5	0

Note. p_c = *2 (i.e., concept 1 and 2).*

p_{uA}*=1 (i.e., concept 3).*

p_{uB}*=1 (i.e., concept 4).*

β *= 9 (i.e., the scale is from 1 to 9. The maximum strength is 9).*

Figure 22 Sample Knowledge Structures for a Project Manager and Developer

Appendix M

Measurement of Cross-understanding

(1) An understanding of knowledge and skills

We would like to learn your perception of your co-worker's knowledge and skills about software development (the co-worker refers to the person we indicated in the mail). Please answer **to what extend do you believe you have a solid understanding of the co-worker's knowledge and skills regarding the following areas**. Please note that the question is NOT about whether your co-work have or doesn't have the knowledge and skills. You can have a solid understanding of your co-worker who doesn't have any knowledge and skills about any of the areas below but you know about his/her knowledge and skills very well (i.e., answer in 7). What we want to know is your understanding of he/she.

	1 I have very little understanding of his/her knowledge and skills	2	3	4	5	6	7 I have extensive understanding of his/her knowledge and skills
1. teamwork skills	☐	☐	☐	☐	☐	☐	☐
2. ability to analyze technical possibilities (e.g., platforms, industry standards, and best practices)	☐	☐	☐	☐	☐	☐	☐
3. skills at developing and managing project life cycles (e.g., monitoring, risk management, verification and validation)	☐	☐	☐	☐	☐	☐	☐
4. knowledge of business strategy	☐	☐	☐	☐	☐	☐	☐
5. negotiation skills (e.g., scope changes, contract)	☐	☐	☐	☐	☐	☐	☐
6. knowledge of IT strategy	☐	☐	☐	☐	☐	☐	☐
7. ability to analyze business environment (e.g., environmental constraints, business functions)	☐	☐	☐	☐	☐	☐	☐
8. technical implementation skills (e.g., programming, testing)	☐	☐	☐	☐	☐	☐	☐

9. customer management skills	☐	☐	☐	☐	☐	☐	☐
10. technical design skills (e.g., architecture, modeling)	☐	☐	☐	☐	☐	☐	☐
11. knowledge of personnel development	☐	☐	☐	☐	☐	☐	☐
12. project planning skills (e.g., work breakdown, estimation)	☐	☐	☐	☐	☐	☐	☐
13. people management skills	☐	☐	☐	☐	☐	☐	☐
14. ability to implement business solution (e.g., changed working practices and attitudes toward the system)	☐	☐	☐	☐	☐	☐	☐
15. knowledge of enterprise design (e.g., the integration of IT and business processes)	☐	☐	☐	☐	☐	☐	☐

(2) An understanding of beliefs

Individuals have different beliefs on what should be valued in or from projects. We would like to learn your understanding of your co-worker's beliefs (the co-worker refers to the person we indicated in the mail). Please answer **to what extend do you think that you have a solid understanding of the co-worker's beliefs regarding the importance of the following aspects of project effectiveness**. Please note that the question is NOT about whether your co-work believe or disbelieve the importance of these project effectiveness criteria. You can have a solid understanding of your co-worker who totally doesn't care about the project effectiveness criteria below but you know about his/her beliefs very well (i.e., answer in 7). What we want to know is your understanding of he/she.

Aspects of project effectiveness	1 I have very little understanding of his/her beliefs	2	3	4	5	6	7 I have extensive understanding of his/her beliefs
1. Beliefs about planning and goal setting	☐	☐	☐	☐	☐	☐	☐
2. Beliefs about flexibility and adaptation	☐	☐	☐	☐	☐	☐	☐
3. Beliefs about utilizing project environment (e.g., resource acquisition)	☐	☐	☐	☐	☐	☐	☐
4. Beliefs about information management and communication	☐	☐	☐	☐	☐	☐	☐
5. Beliefs about values of human resources	☐	☐	☐	☐	☐	☐	☐
6. Beliefs about productivity	☐	☐	☐	☐	☐	☐	☐
7. Beliefs about product and service quality	☐	☐	☐	☐	☐	☐	☐
8. Beliefs about growth of organizations	☐	☐	☐	☐	☐	☐	☐
9. Beliefs about control	☐	☐	☐	☐	☐	☐	☐
10. Beliefs about training and development	☐	☐	☐	☐	☐	☐	☐
11. Beliefs about efficiency	☐	☐	☐	☐	☐	☐	☐
12. Beliefs about morale	☐	☐	☐	☐	☐	☐	☐
13. Beliefs about readiness for changes	☐	☐	☐	☐	☐	☐	☐
14. Beliefs about external entity evaluation (e.g., loyalty to , confidence in, and support given to the project by such external entities as vendors, customers, and users)	☐	☐	☐	☐	☐	☐	☐

15. Beliefs about stability	☐	☐	☐	☐	☐	☐	☐
16. Beliefs about cohesion	☐	☐	☐	☐	☐	☐	☐
17. Beliefs about profitability	☐	☐	☐	☐	☐	☐	☐

Appendix N

Measurements for Task Interdependence and Work History

We would like to learn a bit about how you and your colleague work together in the project.

Please think specifically the colleague and the project we indicate in the email we sent you.

	1 Strongly Disagree	2	3	4 Neutral	5	6	7 Strongly Agree
1. I have to obtain information and advice from my colleague to complete my work.	☐	☐	☐	☐	☐	☐	☐
2. I depend on my colleagues for the completion of my work	☐	☐	☐	☐	☐	☐	☐
3. I have a one-person job and rarely have to check or work with others.	☐	☐	☐	☐	☐	☐	☐
4. I have to work closely with my colleagues to do my work properly.	☐	☐	☐	☐	☐	☐	☐
5. In order to complete their work, my colleagues have to obtain information and advice from me.	☐	☐	☐	☐	☐	☐	☐

6. Please indicate the number of projects that you and your colleague work together: __________
7. Please indicate the number of years that you and your colleague work together: __________

Questions	Measure	Source
Q1-Q5	Task interdependence (Cronbach α: 0.75)	van der Vegt et al. (2000)

Van Der Vegt, G., Emans, B., & Van De Vliert, E. (2000). Team members' affective responses to patterns of intragroup interdependence and job complexity. *Journal of Management*, *26*(4), 633–655.

Appendix O

An Example of ISDP Mental Model

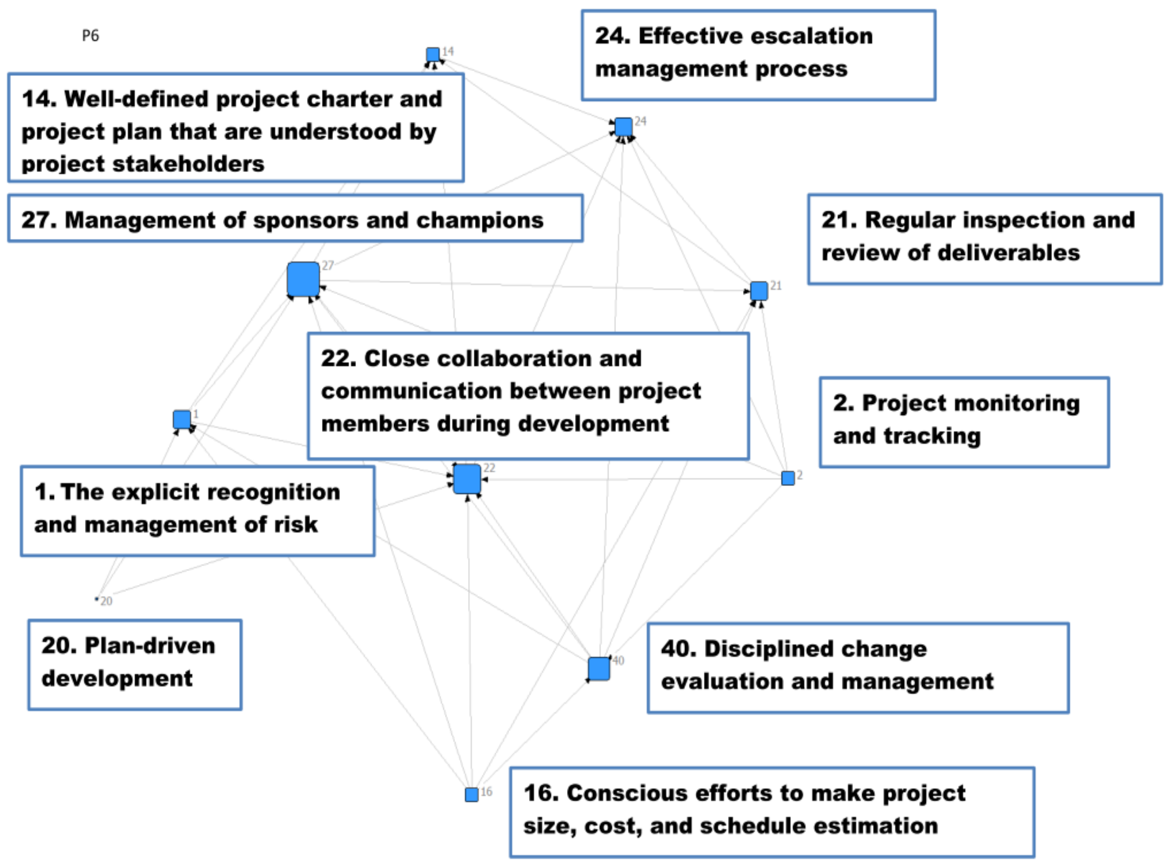


Figure 23 An Example of ISDP Mental Model

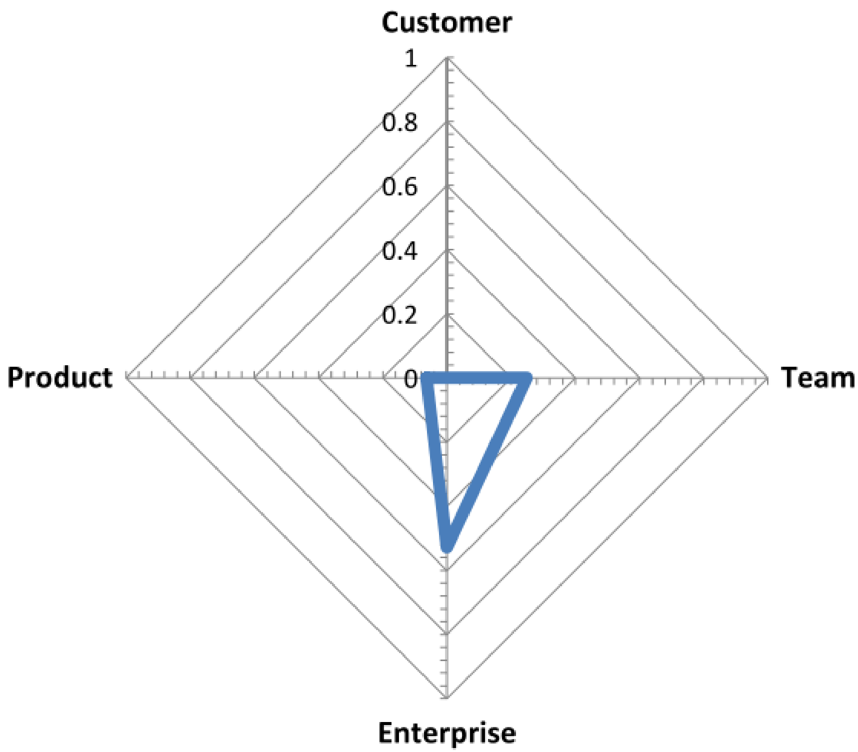


Figure 24 An Example of Belief Orientation

Appendix P

Interview Protocols

(1) To begin, I'd like to learn about your software development experience. Can you briefly describe your experience and expertise relevant to software development?
    a. I am particularly interested in your beliefs about the management of software development projects. When you work on software development project, which types of outcomes do you think are most important for you?
        i. (probe) any other individual, project, team outcomes or organization outcomes you care?
    b. How do you achieve these outcomes?
        i. (probe) What principles you believe? What development/management practices do you adopt?
    c. Which software development principles and practices do you like?'
    d. Which software development principles and practices do you dislike?'

(2) Let's talk a bit about your co-worker.
    a. What do you know about his/her experience and expertise in software development?
    b. Based on your knowledge of him/her, what aspects of outcomes do you think that he/she values deeply?
    c. How does he/she achieve these outcomes?
        i. (prob) What principles he/she believes? What development/management practices he/she adopted?
    d. Which software development principles and practices does he/she like?
    e. Which software development principles and practices does he/she dislike?
    f. Do you think that you know your project manager (or developer) well?
        i. If yes, in what aspects?
        ii. Do you think your understanding of him/her help your collaboration?
            1. If yes, (original question) how does your understanding of him/her help collaboration?
            2. If no, why an understanding isn't an important factor?
        iii. If no, why a lack of understanding?
            1. Would a lack of understanding hinder your collaboration?
    g. Do you think that you and your co-worker hold similar beliefs and preferences regarding the management of software development projects?
        i. If yes, in what aspects?
            1. How do similarities influence your collaboration?

ii. If no, what are major differences

1. Do you think the difference affects your collaboration?

(3) Let's move to your work relationships with your co-worker. By work relationships, I mean whether you and your co-worker work effectively.

a. How would you describe your working experience with your co-worker?

i. If you compare your work relationships with XXX with your work relationships with other co-workers? Would you say it's average, above average, or below average?

b. Give me an example(s) of when your project manager (or developer) and you worked well.

a. Give me an example(s) of when your project manager (or developer) and you had conflict (any disagreements?)

b. How has the work relationship changed since the project kick-off?

i. What caused the changes?

c. Why you and your co-worker have effective (or ineffective) work relationship?

d. In general, what factors influence work relationships?

Appendix Q

Coding Schema for Study 2

Code	Definition	Subcode
Similarity of ISDP mental models	The project manager and developer hold similar knowledge (similarity index > .25) and dominant beliefs.	
Understanding of ISDP mental models	The project manager-developer dyad possesses an accurate understanding of the mental models of one another, including knowledge and beliefs. The accuracy is contingent upon one can accurately describe part of his/her co-worker's ISDP concepts and beliefs.	
Understanding of expertise	Individual possess an accurate understanding of the expertise of his/her co-worker. The expertise in ISD projects is divided into the following categories: planning and control, general management, leadership, communication, team development, client management, system development, problem solving, planning and control, problem solving, and technical expertise.	(1) Client management: the ability to successfully relate to clients during all phases of the project (Napier et al., 2009). (2) Communication: the ability of the IT PMs to effectively speak, write and listen to secure resources, enhance coordination, and ensure that work is completed (Napier et al., 2009). (3) General management: the ability encompasses business and interpersonal skills required to appropriately manage themselves and others (Napier et al., 2009). (4) Leadership: the ability to form and

		communicate a message about the future direction of the project in a way that garners enthusiasm and commitment from others (Napier et al., 2009). (5) Planning and control: the ability of planning, monitoring and controlling project tasks to ensure that the project is completed on time and within the budget (Napier et al., 2009). (6) Problem solving: the ability to address problems efficiently and effectively (Napier et al., 2009). (7) System development: the ability to understand and manage the technical aspects of developing complex, technical systems while controlling for quality. It also involves being able to effectively manage the complexity of creating IT systems. The system development expertise goes beyond technical expertise(Napier et al., 2009). (8) Team development: the ability to create a productive team environment for those working on the project while demonstrating concern for their personal and professional growth (Napier et al., 2009). (9) Technical expertise: the ability to apply technologies to implement the IS application – computer hardware, operating systems and other systems

		software, data base packages and other middleware, etc. (Iivari et al., 2004)
Understanding of needs and constraints	Knowledge about the co-worker's personal aspirations, goals, and personal issues.	
Understanding of social aspects	Knowledge about the co-worker's beliefs, preferences, and sensitive in the non-work related setting.	
Implicit coordination	When the dyad anticipates the actions and needs of one another and task demands. Then, either party dynamically adjusts behaviours accordingly (Rico, Sánchez-Manzanares, Gil, & Gibson, 2008)	(1) Unsolicited project-relevant information: An individual provides his/her co-worker with project-relevant and timely information without having been asked to do so (Entin & Serfaty, 1999) (2) Changes in project process: An individual adjusts the processes of ISD that are aligned with his/her co-worker's abilities, beliefs, and preferences. The process of ISD could be formal in the sense that the processes are formally implemented in projects. It also can be the informal processes in which the dyad has a shared understanding of what should go through to obtain the outcomes. (3) Task assignments: An individual would be able to assign tasks that meet his/her co-worker's abilities, beliefs, and preferences.
Interpersonal mode	The interpersonal mode of explicit	(1) Information exchange: Exchange of

of explicit coordination	coordination depends on communication between co-workers to manage the interdependencies between tasks and people.	information between co-workers in order to manage tasks and people in ISD projects. (2) Feedback seeking and giving: Giving, seeking, and receiving information between co-workers in order to manage tasks and people in ISD projects. (3) Knowledge integration: Integration of diverse expertise that is possessed by co-workers in order to manage tasks and people in ISD projects.
Impersonal mode of explicit coordination	The impersonal mode of explicit coordination requires minimal verbal communication and takes the form of administrative coordination or programming (e.g. formal policies and procedures, project milestones and delivery schedules, project documents and memos, regularly scheduled team meetings, requirement review meetings, or design inspections in software development teams) (Ellwart, 2011).	
Frequent interaction	Frequent and close interaction, formal or informal, between co-workers.	
Mentorship	An experienced person who advises and helps a less experienced person	
Anchoring events	Memory episodes that contains extreme emotional and instrumental	

	content for the dyadic partners (Ballinger & Rockmann, 2010).	
Compatibility between mental models and the impersonal mode of explicit coordination	Individuals show their understanding of the value of existing coordination mechanisms and believe that coordination mechanisms help with coordination effectiveness.	
Interpersonal citizenship behaviors	Individuals demonstrate helping behaviors for their co-worker beyond their job requirements.	
Benevolence trust	The extent to which a trustee is believed to want to do good for the trustor, apart from any profit motives, with synonyms including loyalty, openness, caring, or supportiveness (Mayer et al., 1995).	
Competence trust	Positive expectations about co-workers' skills, competencies, and characteristics for some specific domains (Mayer et al., 1995).	

Appendix R

Study 1 GREB Research Approval

October 19, 2011

Mr. Yi-Te Chiu
Ph.D. Candidate
Queen's School of Business
Queen's University
Kingston, ON K7L 3N6

GREB Ref #: GBUS-324-11; Romeo # 6006317
Title: "GBUS-324-11 An Exploration of Information System Development Mental Models"

Dear Mr. Chiu:

The General Research Ethics Board (GREB), by means of a delegated board review, has cleared your proposal entitled **"GBUS-324-11 An Exploration of Information System Development Mental Models"** for ethical compliance with the Tri-Council Guidelines (TCPS) and Queen's ethics policies. In accordance with the Tri-Council Guidelines (article D.1.6) and Senate Terms of Reference (article G), your project has been cleared for one year. At the end of each year, the GREB will ask if your project has been completed and if not, what changes have occurred or will occur in the next year.

You are reminded of your obligation to advise the GREB, with a copy to your unit REB, of any adverse event(s) that occur during this one year period (access this form at https://eservices.queensu.ca/romeo_researcher/ and click Events - GREB Adverse Event Report). An adverse event includes, but is not limited to, a complaint, a change or unexpected event that alters the level of risk for the researcher or participants or situation that requires a substantial change in approach to a participant(s). You are also advised that all adverse events must be reported to the GREB within 48 hours.

You are also reminded that all changes that might affect human participants must be cleared by the GREB. For example you must report changes to the level of risk, applicant characteristics, and implementation of new procedures. To make an amendment, access the application at https://eservices.queensu.ca/romeo_researcher/ and click Events - GREB Amendment to Approved Study Form. These changes will automatically be sent to the Ethics Coordinator, Gail Irving, at the Office of Research Services or irvingg@queensu.ca for further review and clearance by the GREB or GREB Chair.

On behalf of the General Research Ethics Board, I wish you continued success in your research.

Yours sincerely,

Joan Stevenson, Ph.D.
Professor and Chair
General Research Ethics Board

cc: Dr. Sandy Staples, Faculty Supervisor and Co-applicant
Dr. Jane Webster, Chair, Unit REB
Amy Marshall, c/o Research Office

Appendix S

Study 2 GREB Research Approval

June 03, 2013

Mr. Yi-Te Chiu
Ph.D. Candidate
Queen's School of Business
Queen's University
Kingston, ON K7L 3N6

GREB Ref #: GBUS-384-13; Romeo # 6008170
Title: "GBUS-384-13 Work Relationships between Software Project Managers and Developers"

Dear Mr. Chiu:

The General Research Ethics Board (GREB), by means of a delegated board review, has cleared your proposal entitled **"GBUS-384-13 Work Relationships between Software Project Managers and Developers"** for ethical compliance with the Tri-Council Guidelines (TCPS) and Queen's ethics policies. In accordance with the Tri-Council Guidelines (article D.1.6) and Senate Terms of Reference (article G), your project has been cleared for one year. At the end of each year, the GREB will ask if your project has been completed and if not, what changes have occurred or will occur in the next year.

You are reminded of your obligation to advise the GREB, with a copy to your unit REB, of any adverse event(s) that occur during this one year period (access this form at https://eservices.queensu.ca/romeo_researcher/ and click Events - GREB Adverse Event Report). An adverse event includes, but is not limited to, a complaint, a change or unexpected event that alters the level of risk for the researcher or participants or situation that requires a substantial change in approach to a participant(s). You are also advised that all adverse events must be reported to the GREB within 48 hours.

You are also reminded that all changes that might affect human participants must be cleared by the GREB. For example you must report changes to the level of risk, applicant characteristics, and implementation of new procedures. To make an amendment, access the application at https://eservices.queensu.ca/romeo_researcher/ and click Events - GREB Amendment to Approved Study Form. These changes will automatically be sent to the Ethics Coordinator, Gail Irving, at the Office of Research Services or irvingg@queensu.ca for further review and clearance by the GREB or GREB Chair.

On behalf of the General Research Ethics Board, I wish you continued success in your research.

Yours sincerely,

John G. Freeman

John Freeman, Ph.D.
Professor and Acting Chair
General Research Ethics Board

cc: Dr. Sandfield Staples, Faculty Supervisor
Dr. Jane Webster, Chair, Unit REB
Amy Marshall, c/o Research Office