INVESTIGATING THE CHANGE PROCESSES IN INFORMATION SYSTEMS REQUIREMENTS

by

Atefeh Taghavi

B.Sc., The University of Tehran, 2009
M.Sc., K. N. Toosi University of Technology, 2011
M.Sc., University of British Columbia, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Business Administration)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

August 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the dissertation entitled:

Investigating the Change Processes in Information Systems Requirements

submitted by Atefeh Taghavi in partial fulfillment of the requirements for

the degree of Doctor of Philosophy

in Business Administration

Examining Committee:

Prof. Carson Woo, Sauder School of Business, UBC
Co-supervisor

Prof. Ning Nan, Sauder School of Business, UBC
Co-supervisor

Prof. Ronald Timothy Cenfetelli, Sauder School of Business, UBC
Supervisory Committee Member

Prof. Hasan Cavusoglu, Sauder School of Business, UBC
University Examiner

Prof. Reid Holmes, Department of Computer Science, UBC
University Examiner

Additional Supervisory Committee Members:

Prof. Victoria Lemieux, School of Information, UBC
Supervisory Committee Member

Abstract

Success of information systems (IS) projects is deeply dependent on the quality of the requirements informing their design. Specifically, if the requirements are not precisely determined, IS cannot meet the business needs of organizations. One of the most enduring challenges in managing requirements is ongoing change in requirements. Changes emerge due to various reasons. The main goal of this dissertation is to shed more light on change drivers and change processes in IS requirements. To achieve this goal, we conduct three separate but interdependent studies.

The first study elicits and synthesizes various change drivers from the extant literature. By employing a design science research methodology, we propose the Socio-Technical Change Framework and Socio-Technical Requirements Change Method. These two artifacts are drawing upon models in socio-technical systems studies. The proposed framework elaborates on how various social and technical change drivers jointly develop the changes in requirements. The proposed method also provides IS analysts with a new solution to anticipate potential future changes in requirements.

The second study investigates the change drivers in contemporary IS projects. We approach this study from a qualitative research methodology. The results from the interviews and surveys reveal 14 categories of change drivers. These change drivers are similar to the change drivers in the first study. However, we explore that the contemporary context includes interesting unique characteristics, which moderate the change processes in IS requirements.

The third study dives more deeply into some of the findings from the second study. Specifically, we examine how two of the important change drivers — changes in the environment and changes in user expectations — and one prominent characteristic of contemporary projects — interdependency between requirements — jointly influence the final changes in requirements. To address this query, we employ an agent-based simulation model. We explore interesting insights into the impact of the interdependency between requirements and learning patterns of the population on requirements change.

In sum, the findings of all three studies contribute to the extant body of knowledge in IS requirements and open interesting avenues for future research. They also have practical implications for IS analysts, project managers, business owners, and third-party vendors.

Lay Summary

Organizations have been investing hugely in their digital technologies. However, many information systems development projects still encounter failure. An important reason is business users continuously change their mind about what information systems should do. In this dissertation, we aim to shed light on the reasons behind the changing needs of organizations. First, we elicit the most important reasons from the existing literature. We propose a framework that integrates all these reasons. Second, we collect new insights into the requirements change from real-world projects. This helps us complement our previous theoretical findings. Third, we employ a computer simulation model to deeply examine how these change reasons interact with each other.

Preface

All aspects of the research work presented in this dissertation have been conducted by Atefeh Taghavi. This research work has significantly benefited from the feedback and revisions provided by members of the supervisory committee.

The research work in Chapter Three has been conducted in accordance with the research ethics at the University of British Columbia. UBC Behavioral Research Ethics Board has approved this research under the certificate number H18-01752.

This dissertation is unpublished.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I feel truly grateful to many people who have kindly supported me during my PhD journey. First, I would like to sincerely thank my wonderful supervisors Dr. Carson Woo and Dr. Ning Nan. Thanks for inspiring, challenging, and guiding me. Without your constant support and encouragement, my PhD success would not have been possible. Carson, you have been such an excellent mentor and role model to me in both research and teaching. It has been an honor to be your student. Thank you very much for patiently training me to become the researcher I am today. I truly appreciate your precious time, presence, attention, and effort during all the past years. Ning, thanks a lot for giving me all those valuable guidance and support. You have shown me the beauty of our complex world! By looking through this new window, I truly enjoy and admire how every little thing can create such amazing outcomes. Moreover, you have always inspired and helped me to become a more creative and independent researcher. Thank you very much.

I would like to express my enduring gratitude to my other two amazing committee members Dr. Ronald Timothy Cenfetelli and Dr. Victoria Lemieux. Ron, many thanks for kindly guiding me, specially through my qualitative study. I truly appreciate your time and effort. Vicki, thanks a lot for all your valuable feedback, support, and kindness.

I am thankful to many great people at Sauder School of Business. My special thanks to Dr. Izak Benbasat, Dr. Yair Wand, and all other faculty in MIS group for their kind help and constructive feedback in the past few years. A big thank you to all my fellow doctoral students, Pat, Mona, Forough, Vibhuti, Ozum, Natalya, Barnini, Dennis, Wenyi, Hossein, Nathan, Ruijing, Dr. Arash, Dr. Amin, Dr. Hongki, Dr. Daniel, Dr. Johannes, … You have always supported me with your positive energy and encouragement. Also, I would like to genuinely thank Ms. Elaine Cho, Ms. Paula Chang, and Ms. Debra Harris for all their valuable help during these years.

I would like to express my sincere appreciation to my other amazing professors, whose teaching has guided me through my doctoral research. Dr. Sandra Mathison and Dr. Olga Volkoff, many thanks for introducing me to the colorful world of qualitative research. Dr. Martin Schulz, thanks a lot for teaching me the essence of organizational learning.

I would like to really thank Sauder School of Business, School of Graduate and Postdoctoral Studies at UBC, Natural Sciences and Engineering Research Council (NSERC) of Canada, and Social Sciences and Humanities Research Council (SSHRC) of Canada for kindly providing me with the financial support to conduct my research. Also, I would like to take this opportunity to sincerely appreciate the generous and volunteer help of all the people who supported me to collect data in my second study. Thanks very much for your valuable time.

I am so grateful and blessed to have wonderful friends who have genuinely supported and uplifted me during my PhD journey. A special thanks to my dearest Dr. Marjan, my beloved sister, who has always kindly encouraged me to be the best I can in every way.

Last but most importantly, I would like to deeply thank my family who have been with me in every moment of my PhD journey. My lovely mom and dad, words cannot express how grateful I am for having you in my life. Many thanks for giving me your unconditional love and supporting me in all aspects of my life. Without your amazing help, I would not be here today. Thank you so very much!

To my parents

Chapter One

Introduction

To successfully meet the business needs of the organization, information systems (IS) have to be designed based on correct and complete requirements of users and other stakeholders in the organization (Xiao et al. 2018; Schneider et al. 2018; Meth et. al 2015; Shuraida and Barki 2013; Appan and Browne 2012; Chakraborty et al. 2010; Zowghi and Gervasi 2004; Davis 1982). Managing the requirements is known as one of the most challenging tasks in IS development projects (Davidson 2002). More specifically, requirements should be precisely identified, negotiated, evaluated, prioritized, documented, and tracked in each project.

As a result, a rich body of literature has been shaped since the 1970's to discuss various social and technical aspects of managing requirements, underline different related concerns and issues, and finally propose effective and interesting solutions to address those challenges and problems. For example, the extant literature in IS requirements has discussed *different requirements elicitation methodologies* (Ramesh et al. 2010; Browne and Rogich 2001; Marakas and Elam 1998; Byrd et al. 1992; Davis 1982; Munro and Davis 1977), *empirical and theoretical studies about the relationships among participants in the requirements elicitation process* (Rosenkranz et al. 2014; Shuraida and Barki 2013; Rosenkranz et al. 2013; Appan and Browne 2012; Jayanth et al. 2011; Chakraborty et al. 2010; Appan and Brown 2010; Pitts and Browne 2004; Urquhart 2001; Sutton 2000; Byrd et al. 1992; Davis 1982), *various studies on agility in IS development* (Kude et al. 2020; Ramesh et al. 2012; Lee and Xia 2010; Ramesh et al. 2010; Port and Bui 2009; Maruping et al. 2009), and so on.

One of the most enduring challenges in managing IS requirements, which has been investigated by many studies, is *ongoing change in requirements* (Schneider et al. 2018; Jayatilleke et al. 2018; McGee and Greer 2012). Requirements are called *moving targets* (Maruping et al. 2009; Davidson 2002). Specifically, in our contemporary world, change appears to be an inevitable characteristic of IS requirements (Jayatilleke et al. 2018; Jarke et al. 2011; Port and Bui 2009). Managing changes in requirements is very important. Particularly, in large and complex systems it is known as a complicated process (Jayatilleke et al. 2018).

Broadly speaking, requirements change because of the two main reasons. First, sometimes requirements are not thoroughly identified and documented at the beginning of a project. As a result, some IS develop based on inaccurate requirements and are unable to successfully fulfill certain business needs. Consequently, requirements must be changed later to address the needs in the organization. This process may happen due to several reasons. Two of the most well-known reasons are the memory limitations of users and communication problems between users and analysts (Rosenkranz et al. 2014; Shuraida and Barki 2013; Rosenkranz et al. 2013; Appan and Browne 2012; Jayanth et al. 2011; Ramesh et al. 2010; Chakraborty et al. 2010; Appan and Brown 2010; Pitts and Browne 2007; Pitts and Browne 2004; Urquhart 2001; Sutton 2000; Byrd et al. 1992; Davis 1982). Generally, this group of requirements changes is considered troublesome for the project managers and business owners, as these changes result in additional significant cost for the organization (Jayatilleke et al. 2018; Meth et. al 2015; McGee and Greer 2012; Maruping et al. 2009; Rolland et al. 2004; Anderson and Felici 2002; Zowghi and Nurmuliani 2002).

Second, IS requirements change as a response to other changes in the dynamic organizational or environmental context in which IS are designed or implemented (Schneider et al. 2018; McGee and Greer 2012; Holmström and Sawyer 2011; Jarke et al. 2011; Maruping et al. 2009; Rolland et al. 2004; Bowen et al. 2002; Sutton 2000; Patel 1999; Yu 1997; Zowghi and Offen 1997; Ecklund et al. 1996; Chung et al. 1996). These changes can emerge as a result of the change in economics, market demands, regulations, organizational strategies, business processes, IS project constraints, and so on. Unlike the first group of requirements changes, the second group is mostly considered as an opportunity to better support the business needs of the organization

The main purpose of this dissertation is to shed light on both groups of changes in IS requirements. More specifically, in this dissertation, we strive to attain the following objectives:

First, we collect different root causes of requirements[1] change from the extant literature (i.e., different factors in the two aforementioned groups) and synthesize them. We propose a comprehensive model that includes all of these factors.

Second, we use this comprehensive model to propose a method. We expect this method can support IS analysts to evaluate the accuracy of the current requirements and help analysts anticipate potential future changes in requirements, as much as possible.

Third, we combine our knowledge drawn upon the literature with empirical insights collected from the real-world. More specifically, we study the root causes of change in requirements in contemporary IS development projects.

Fourth, we investigate how various change reasons can interact with each other. Most often, more than one root cause of requirements change exists in IS development projects (e.g., changes in the business processes and communication problems between IS analysts and users). We examine the joint influence of various root causes of change on the final changes in requirements.

To accomplish these objectives, we conduct three studies. These studies are presented in Chapter Two, Chapter Three, and Chapter Four of this dissertation.

In **Chapter Two**, we conduct a comprehensive literature review to investigate the *"change drivers"* — another term we use in this dissertation to refer to the root causes of change in requirements. These change drivers are elicited from the most recognized journals within IS and requirements engineering disciplines. As elaborated in this chapter, the extant literature has identified a broad spectrum of change drivers, ranging from social change drivers (e.g., new market demands, regulations, business goals) to technical ones (e.g., change in available technologies on the market, technical capacities within organizations, and technology features used in IS projects).

However, the existing studies do not reveal further information on how these change drivers are interconnected. We approach this research from a *design science research*

[1] In this dissertation, by the term "IS requirements" or simply "requirements", we refer to the business requirements of IS users and other stakeholders in the organization (e.g., business owners and project managers). Requirements express what should be accomplished through IS to support the business. Different technologies are included in IS. However, IS requirements are more abstract (and have a broader scope) than the technology requirements. After collecting and modeling the precise list of IS requirements, they will be translated to specific technical requirements, which will be employed later by software/hardware developers and programmers.

methodology. Drawing upon the existing *socio-technical systems* studies, we propose the *"Socio-Technical Change Framework"*. Our framework models how different social and technical change drivers in the environment, organization, and IS development project are interconnected.

Finally, drawn from this framework, we propose the *"Socio-Technical Requirements Change Method"*. This method supports IS analysts to evaluate the accuracy of the current requirements and anticipate potential future changes in them.

In **Chapter Three**, we employ the *qualitative research methodology* to obtain empirical insights into requirements changes in contemporary IS development projects. More specifically, we set our lens on off-the-shelf packaged solutions, including both on-premise and cloud-based systems. We also collect data from IS projects which utilize the most popular and recent technologies (e.g., mobile applications, machine learning, and data analytics). We focus on contemporary IS development projects because the existing studies provide little explanation on the impact of modern context on requirements change. We investigate whether change drivers in the modern context are different from change drivers collected by our former study.

In this chapter, we present the results of our *interviews* with IS consultants. We identify 14 categories of change drivers. Also, we discuss how the modern context moderates the impact of these change drivers. To increase the validity of our results, we also use another source of data from *surveys*. The findings from the surveys confirm all the 14 change drivers and shed more light on the moderating impacts of the modern context.

In **Chapter Four**, we study the joint influence of two change drivers and one prominent characteristic of off-the-shelf solutions on IS requirements change. These three factors all draw upon the results of Chapter Three. More specifically, in this research, we investigate how "changes in the technologies available on the market", "changes in users' expectations", and "level of interdependency between requirements of IS" impact changes in requirements.

In our study, we employ an *agent-based simulation model* to represent the interactions between IS users, the environment, and the organization. After implementing a base model, we conduct two experiments to explore the impact of the level of interdependency between requirements and the "pattern of learning of IS users" on requirements change. As will be introduced in this chapter, the pattern of learning is an essential construct in the process of changing the users' expectations. We explore interesting findings, which are all discussed in this chapter. Finally, based on the results of the experiments, our study proposes several theoretical propositions that could guide further research works.

Lastly, in **Chapter Five**, we conclude this dissertation by reflecting on the main findings from the previous chapters and proposing directions for future studies.

Chapter Two

An Analysis of Changes in Information Systems Requirements: A Socio-Technical Systems Approach

Introduction

Success of IS projects highly depends on the quality of the requirements — particularly correctness, completeness, and consistency level of requirements (Xiao et al. 2018; Schneider et al. 2018; Meth et. al 2015; Shuraida and Barki 2013; Appan and Browne 2012; Chakraborty et al. 2010; Zowghi and Gervasi 2004; Davis 1982). In other words, lack of accuracy in requirements results in lower reliability of IS (Bowen et al. 2002) and project failure (Schneider et al. 2018; Xiao et al. 2018). Over the past few decades, a large number of IS studies have raised various insights into the process of requirements management. These studies underline different problems and challenges and propose solutions to overcome the complications. The extant body of knowledge in IS requirements sheds light on a wide spectrum of topics, including but not limited to the *project management aspects of requirements* (Moe et al. 2017; Maruping et al. 2009; Guinan et al. 1998), *various methodologies to elicit the requirements* (Ramesh et al. 2010; Browne and Rogich 2001; Marakas and Elam 1998; Byrd et al. 1992; Davis 1982; Munro and Davis 1977), *behavioral and communicational aspects of the process of requirements determination* (Rosenkranz et al 2014; Appan and Browne 2012; Jayanth et al. 2011; Pitts and Browne 2004), and so on.

One of the most enduring challenges in the management of IS requirements is *ongoing change in requirements*. From one perspective, changes need to be prevented. They are considered unfortunate and troublesome, because they result in additional significant costs for the organizations (Jayatilleke et al. 2018; Meth et. al 2015; McGee and Greer 2012; Maruping et al. 2009; Rolland et al. 2004; Anderson and Felici 2002; Zowghi and Nurmuliani 2002). Particularly, changes that emerge after the implementation phase of a project are very expensive. As elaborated by Boehm and Basili (2001), *"finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase"* (p. 135). A majority of these changes occur as a result of project management issues, communication problems between IS analysts and users, and employing improper techniques to elicit, analyze, or validate the requirements. However, from a more positive perspective, changes in requirements should be supported. They are considered as innovative opportunities to better address the business needs of organizations. These changes often emerge as a response to changing needs of organizations and the business environment (Schneider et al. 2018; McGee and Greer 2012; Holmström and Sawyer 2011; Jarke et al. 2011; Maruping et al. 2009; Bowen et al. 2002).

Regarding both aforementioned perspectives, the main purpose of this study is to deeply investigate the root causes of change and model various change processes in IS requirements[2]. We are also interested in proposing a method that can help IS analysts anticipate potential changes in requirements, as much as possible.

Existing literature identifies a large number of root causes of change. These root causes range from social factors (e.g., market demands, regulations, organizational goals, and project management constraints) to more technical ones (e.g., available technologies on the market, technical capacities, and technology policies in organizations). These findings are indeed very significant. However, they do not reveal further information about how various social and technical factors are interconnected to each other, nor how they can jointly develop the changes in IS requirements.

To address this gap, our research employs some models from *socio-technical systems (STS)* studies. The STS approach was originally introduced by organizational theorists to recognize both the social and technical needs of organizations. It later evolved and was operationalized by various IS research works, particularly in the areas of IS design (Mumford 2000, 1997; Mumford and Weir 1979; Bostrom and Heinen 1977a, 1977b) and IS change (Lyytinen and Newman 2008; Lee and Xia 2005). STS models provide us with useful insights into the relationship between various social and technical change drivers.

---

[2] In the context of IS requirements, we define a "*change process*" as *a series of steps that start from a root cause of change and end with the actual requirements change*. In this dissertation, we also use the term "*change driver*" to refer to any root cause of change in IS requirements.

In this study, we address the following research questions: ***How are social and technical change drivers interconnected to each other? How do they jointly form the change processes in IS requirements? How can we utilize STS models to elaborate on the change processes? How this knowledge can be used to anticipate, and mitigate the negative effects of, potential changes in requirements?***

To answer these questions, we apply a *design science research methodology*. Drawing upon existing STS models and requirements change studies, we propose the ***Socio-Technical Change Framework***. This framework elaborates on how various social and technical change drivers in the environment, organizations, and IS development projects are interconnected to each other, and how they jointly shape different change processes in requirements. Drawing from the results of the Socio-Technical Change Framework, the ***Socio-Technical Requirements Change Method*** is presented next. This method enables IS analysts to analyze the potential consequences of different change drivers and anticipate further changes in requirements.

The remainder of this chapter is organized as follows. In the next section, we elicit and synthesize the change drivers of requirements from the recognized research works within the IS requirements literature. Next, we review the related STS models. We then propose the Socio-Technical Change Framework and Socio-Technical Requirements Change Method. Finally, we conclude this research with a discussion of the main implications of these two artifacts and future research directions.

A Literature Review on the Changes in IS Requirements

Rolland et al. (2004) define requirements change as a set of gaps between the requirements specification of the current system (*As-Is map*) and the requirements specification of the future system (*To-Be map*). These changes usually emerge in any form of the requirements addition, requirements deletion, or requirements modification (Nurmuliani et al. 2004).

In many cases, requirements are changed because they were originally missed, or they were not precisely identified by clients[3] and IS analysts at the beginning of a project

[3] In this dissertation, different terms will be used to describe different members of the client team. To avoid ambiguity for readers, we explain these terms here. Besides the "business owners" and "managers", we talk about "end-users", "representatives", and "subject matter experts". End-users, or simply "users", are

(Holmström and Sawyer 2011). This might be an outcome of project management issues, communicational and behavioral problems, employing improper techniques to analyze and validate the requirements, and so on. For example, the communicational and behavioral problems can happen due to lack of involvement of users, memory limitations of users, and misleading behavior of IS analysts (Rosenkranz et al. 2014; Shuraida and Barki 2013; Rosenkranz et al. 2013; Appan and Browne 2012; Jayanth et al. 2011; Ramesh et al. 2010; Chakraborty et al. 2010; Appan and Brown 2010; Pitts and Browne 2007; Pitts and Browne 2004; Urquhart 2001; Sutton 2000; Byrd et al. 1992; Davis 1982). In these cases, the intended IS are designed based on an incomplete and incorrect set of requirements, and consequently they will not be able to support all the business needs of clients in the organization. As a solution, the requirements should be modified later to address the overlooked needs. Existing requirements literature proposes various strategies and techniques to minimize the communicational and behavioral challenges (for example, see Vitharana 2016; Jayanth et al. 2011; Zappavigna and Patrick 2010; Pitts and Browne 2007; Pitts and Browne 2004; Grünbacher et al.; Marakas and Elam 1998).

In addition to challenges for identifying and specifying accurate requirements at the beginning of a project, requirements are often prone to future changes. Requirements should be changed as the organizational or environmental context in which the IS are designed or implemented evolve over time (Schneider et al. 2018; McGee and Greer 2012; Holmström and Sawyer 2011; Jarke et al. 2011; Maruping et al. 2009; Rolland et al. 2004; Bowen et al. 2002; Sutton 2000; Patel 1999; Yu 1997; Zowghi and Offen 1997; Ecklund et al. 1996; Chung et al. 1996). In these cases, requirements need to evolve to support ongoing changes in the surrounding context.

Recognizing the dynamic nature of requirements is one of the fundamental principles in our contemporary requirements research and practices (Jarke et al. 2011). It is essential to investigate the root causes of change and explore the change process in IS requirements. This knowledge not only enables IS analysts to improve the quality of the requirements they

---

organizational employees who should directly deal with the system in the future. Since it is difficult for all the end-users to attend the requirements gathering sessions (particularly in large size projects), a group of representatives of users are selected to identify the requirements of all users on behalf of them. Subject matter experts are organizational employees who possess special knowledge and skills about important aspects of the business and help IS consultants to precisely understand the business domains. The term "client" refers to all of these members.

collect, but it also supports them to take steps forward in anticipating potential requirements changes that can occur in the future (McGee and Greer 2012). We have reviewed the existing literature in IS requirements that identify the root causes of change. We have mainly focused on the recognized journals within IS and requirements engineering disciplines, including *Management Information Systems Quarterly*, *Information Systems Research*, *Journal of Association for Information Systems*, *Journal of Management Information Systems*, *European Journal of Information Systems*, *Journal of Information Technology*, *Information Systems Journal*, *Journal of Strategic Information Systems*, and *Requirements Engineering.* In addition to the articles published in the journals mentioned above, we also have conducted a broad search within the *Google Scholars* database. The details of the search processes are documented in Table 2.1[4].

Table 2.1 Sources of the Literature Review

Journal Name	**Search and Selection Process**
Management Information Systems Quarterly	• Keywords: "Requirements" as a part of the title • Publication years: After 2000 • Results: 2 most relevant articles were selected out of 2. 1 article explicitly identifies the root causes of change in requirements.
Information Systems Research	• Keywords: "Requirements" as a part of the title • Publication years: After 2000 • Results: 2 most relevant articles were selected out of 3. 1 article explicitly identifies the root causes of change in requirements.
Journal of Association for Information Systems	• Keywords: "Requirements" as a part of the title • Publication years: After 2000 • Results: 7 most relevant articles were selected out of 8. 3 articles explicitly identify the root causes of change in requirements.
Journal of Management Information Systems	• Search within all the keyword indexes including the "requirements" word

[4] We performed our last search in December 2019. We focused on the studies published after 2000. The only exception was the *Requirements Engineering* journal. Since the *Requirements Engineering* journal was launched in 1996, we searched the whole database of this journal.

Regarding the search and selection process, we attempted to follow the same search procedure for all the journals. However, for the *Journal of Management Information Systems,* we needed to check its list of keywords instead of using our own keywords. The reason was due to a slightly different search mechanism in the online archive of this journal. In addition, regarding the *Requirements Engineering* journal, we needed to focus on the "change" or "evolution" keywords to obtain more specific results.

Journal Name	Search and Selection Process
	• Publication years: After 2000 • Results: 5 most relevant articles were selected out of 5. 1 article explicitly identifies the root causes of change in requirements.
European Journal of Information Systems	• Keywords: "Requirements" as a part of the title • Publication years: After 2000 • Results: 7 most relevant articles were selected out of 11. 4 articles explicitly identify the root causes of change in requirements.
Journal of Information Technology	• Keywords: "Requirements" as a part of the title • Publication years: After 2000 • Results: 2 most relevant articles were selected out of 2. 2 articles explicitly identify the root causes of change in requirements.
Information Systems Journal	• Keywords: "Requirements" as a part of the title • Publication years: After 2000 • Results: 5 most relevant articles were selected out of 9. 2 articles explicitly identify the root causes of change in requirements.
Journal of Strategic Information Systems	• Keywords: "Requirements" as a part of the title • Publication years: After 2000 • Results: 3 most relevant articles were selected out of 12. 1 article explicitly identifies the root causes of change in requirements.
Requirements Engineering	• Keywords: "Change" or "Evolve" or "Evolution" as a part of the title • Publication years: After 1996 (from the first volume) • Results: Out of 12 article, 5 pieces explicitly identify the root causes of change in requirements.
Google Scholars Database	• Keywords: "information systems, requirements, change"; "systems, requirements, change"; "socio-technical, requirements engineering" • Publication years: No limitation • Results: By excluding the articles found from the 8 journals above, we selected 4 new pieces that explicitly identify the root causes of change in requirements.

We reviewed all the collected articles. If an article explicitly mentioned at least one reason for changes in requirements, then we selected that piece as one of our references. In total, our literature review yielded a sample of 24 studies that explicitly elaborate on various potential drivers of change in IS requirements. The results of this literature review are summarized in Table 2.2.

We find that existing studies have identified a wide spectrum of change drivers ranging from social factors, such as *competition*, *goals*, *policies*, and *culture* to more technical ones, such as *available technologies*, *technical capacities in an organization*, and *capabilities of different technologies*.

Table 2.2 Requirements Change Drivers Collected from the Existing Literature

Change Category	Change Driver	References
Changes in the environment and market	Market demand and attitude towards a technology	Land (1982), Ecklund (1996), Nurmuliani et al. (2004), Mathiassen et al. (2007), Chakraborty et al. (2010), Jarke et al. (2011), McGee and Greer (2012), Schneider et al. (2018), Xiao et al. (2018)
	Competition	Rolland et al. (2004), Ramesh et al. (2010)
	Economics	Land (1982)
	Regulations and laws	Land (1982), Ecklund (1996), Jarke et al. (2011), McGee and Greer (2012).
	Available technologies	Ramesh et al. (2010)
	Availability of related systems (e.g. plugins)	Schneider et al. (2018)
Changes in the organization	Business goals and strategies	Patel (1999), Davidson (2002), Nurmuliani et al. (2004), Rolland et al. (2004), Schneider et al. (2018)
	Policies	Ecklund (1996), Chung et al. (1996), Patel (1999), Nurmuliani et al. (2004), Jarke et al. (2011), McGee and Greer (2012), Schneider et al. (2018), Kirsch and Haney (2006)

Change Category	Change Driver		References
	Culture	Values	Tuunanen and Kuo (2014)
		Power and politics	Bergman et al. (2002), McGee and Greer (2012).
		Management style and control systems	Land (1982), Patel (1999)
	Structure		Land (1982)
	Rules		Jayatilleke et al. (2018)
	Business Processes		Ecklund (1996), Patel (1999), McGee and Greer (2012), Schneider et al. (2018)
	Job descriptions and roles		Patel (1999)
	Capacities and support of internal technical team		Schneider et al. (2018)
	Technology policies		Schneider et al. (2018)
	Business value of technology and technology-enabled work practices		Davidson (2002), Schneider et al. (2018)
Changes in the IS development project	Scope and goals		Nurmuliani et al. (2004), Kirsch and Haney (2006), Ramesh et al. (2010), Holmström and Sawyer (2011)
	Complexity level of the problem under analysis		Hickey and Davis (2004)
	Management and culture		Thanasankit (2002), Hickey and Davis (2004), Kirsch and Haney (2006)
	Cost and time		Hickey and Davis (2004), McGee and Greer (2012), Schneider et al. (2018)
	Project participants, their expectations and experience		Davidson (2002), Hickey and Davis (2004), Mathiassen et al. (2007), Hanisch and Corbitt (2007), Maruping et al. (2009), Ramesh et al. (2010),

Change Category	Change Driver	References
		Rosenkranz et al. (2014), Schneider et al. (2018)
	Skills of analysts and developers	Hickey and Davis (2004)
	Requirements elicitation technique	Hickey and Davis (2004)
	Communication between analysts and users	Holmström and Sawyer (2011)
	Delivery strategies	Davidson (2002), Hickey and Davis (2004)
	Challenges during development and programming	Ramesh et al. (2010)
	Technology type and capabilities	Davidson (2002), Hickey and Davis (2004), McGee and Greer (2012)

We also imply that these change drivers emerge from different contexts. For example, while some of them emerge from the organization, some others are drawn from the environment. In general, we have come up with three main categories of contexts: *environmental and market context*, *organizational context*, and *IS development projects context*. The first category includes the social or technical change drivers that emerge from the environment (e.g., new government regulations or technologies available on the market). The second category contains the social or technical change drivers initiated by the organization (e.g., new organizational goals, culture, or technical capacities). Finally, the third category includes the social or technical change drivers that emerge from the IS development project (e.g., new decisions about the timeline, budget, or system delivery strategies).

For the purpose of clarification, we should note that these categories have emerged from our analysis of the change drivers (similar to the coding procedure in qualitative research methodology). They helped us build a preliminary mental model of a hierarchy of change drivers (e.g., to see which change drivers are more similar to each other than to those in other categories). After reviewing the STS models, we found these categories are very similar to some of the concepts in the STS models. Consequently, they later officially were integrated into our framework. This will be elaborated in the next sections.

Each of the references in Table 2.2 presents one or more root cause(s) of change in requirements. Their recognition is indeed a huge contribution to the IS requirements body of literature. However, these studies do not reveal further information on how the changes in various social and technical change drivers result in the final change in IS requirements. It is suspected that some interconnections exist between the aforementioned change drivers. To address this gap, this study investigates how various social and technical change drivers are connected to each other, and how they jointly develop the change processes in IS requirements. To tackle this query, we look at change processes from the STS perspective. We employ the concepts and processes introduced by STS studies as the main theoretical angle in our research.

Socio-Technical Systems Perspective

Background in Organization and IS Studies

The STS perspective originally emerged as a psychology technique by a group of researchers and therapists, at *Tavistock Institute of Human Relations of London*. This technique was used to help war-damaged soldiers recover their psychological health after the Second World War (Mumford 2006). At that time, many of these people had to perform very structured and routine tasks as the main parts of their job, resulting in no opportunity for learning, personal development, social support, and job satisfaction for them. The original STS approach proposed that, when designing organizations and jobs, organizational designers should give an equal weight to both technical and human factors whenever possible (Mumford 2006). We should emphasize that, in those years, the definition of technology mainly included machines, equipment, tasks, and operations, which were used to improve the productivity and reduce costs (Mumford 2006; Bostrom and Heinen 1977a). Over time, STS researchers have extended their theories and introduced a set of principles to improve the STS design for organizations (for example, see Cherns (1976)).

IS literature has widely applied and extended STS theories and models from the beginning of the management information systems field, particularly in IS design studies. A majority of these studies consider any organization as a socio-technical system, which is composed of two interconnected subsystems — social and technical. These studies define the main purpose of IS design as to create a system that can support both social and technical needs in an organization. In the first volume of *MIS Quarterly* journal, Bostrom and Heinen have published two consecutive articles about the socio-technical design. These two pieces explain that the main reason for IS failures is the lack of an adequate socio-technical design approach for IS (Bostrom and Heinen 1977a, 1977b). They suggest that IS analysts and designers think about IS as intervention strategies to promote the organizations from both social aspects (e.g., quality of working life) and technical aspects (e.g., productivity in the organization).

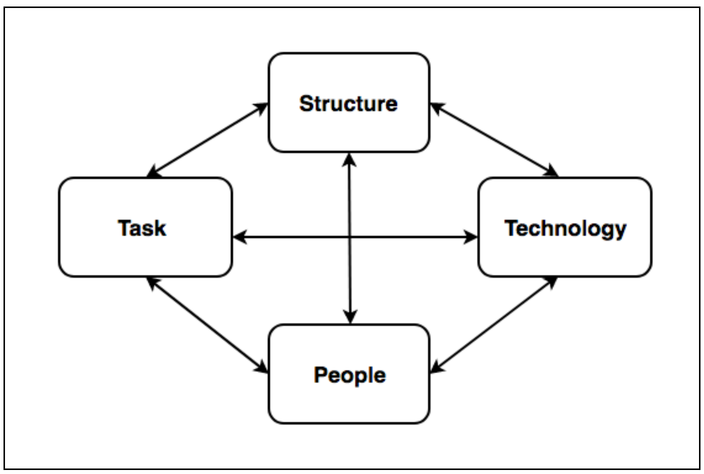
ETHICS (standing for *effective technical and human design of computer-based systems*) is another well-known design method from a STS perspective (Mumford 2000, 1997; Mumford and Weir 1979). User participation is an essential aspect of ETHICS design. ETHICS provides IS analysts and designers with a set of guidelines to precisely describe the existing work systems in the organization, assess the current level of job satisfaction and efficiency — as two key measures of social and technical success in the organization —, determine socio-technical goals, and finally identify solutions to improve the existing work systems.

Work system method (Alter 2006, 2008, 2013) expands upon the previous methods from various angles. Most importantly, it precisely elaborates on what a work system is. Although this term is a central concept in STS studies, it was not precisely defined before (Alter 2013). A work system is a system *"in which human participants and/or machines perform work (processes and activities) using information, technology, and other resources to produce specific products/services for specific internal and/or external customers"* (Alter 2013, p. 75). Similar to the other two methods, but in a more detailed fashion, the work system method analyzes the current work system, measures several key performance indicators for both social and technical subsystems, and identifies the elements that need further improvement. Work systems are viewed as STS in general. However, Alter (2013) emphasizes that they also cover completely automated systems, without the intervention of a human being, which is beyond the scope of classic STS.

Application of STS Approach in Studying IS Change and Requirements

In addition to the aforementioned works, some IS studies explicitly consider IS as socio-technical systems and analyze various social and technical components in them. One of these studies, which we use as our influential theoretical reference here, is the *punctuated socio-technical change model of IS* (Lyytinen and Newman 2008). Drawing upon *Leavitt's change model of organizations* (1964), this study investigates how any misalignment within and between various social and technical components in the IS and associated environment shapes a dynamic change model in IS. Figure 2.1 demonstrates Leavitt's diamond model of organizational change. This model views organizations as systems that are composed of four interacting pieces: *structure*, *people (or actors)*, *task*, and *technology*. Changes in any of these elements could have direct/indirect change effects on all the other three components.

Figure 2.1 Leavitt's Change Model of Organizations (1964)



Lyytinen and Newman (2008) considers IS change as a multi-level change. This model has three interconnected levels — the building system, work system, and organizational environment. Each level is a separate socio-technical system, which includes the actors, structure, task, and technology components. According to this model, IS change should be studied both vertically and horizontally between and within three levels.

Whenever there is a misalignment between the social or technical components of each system (e.g., due to learning, malfunctioning, and replacement), the system moves from a stable state to an unstable state. Consequently, as a response to the misalignment, either other social and technical components within the same system adapt to the new situation incrementally, or the system rewrites its composition rules. In either case, the system should move to a stable state.

Later, Sarnikar et al. (2014) apply the punctuated socio-technical change model of IS into the requirements elicitation context (which is not captured by Lyytinen and Newman's study (2008)), and present the *socio-technical requirement elicitation process model*. This model illustrates that IS requirements emerge when there is a misalignment or gap between the social and technical components of an organization. For example, if an actor's knowledge and skills are not appropriate for completing a specific task (misalignment between actors and task components), then a new requirement should emerge to address the mentioned problem. As a potential requirement, for example, the intended system should provide the actor with the required knowledge to accomplish her task. Although this research moves closer to digesting the change drivers, it only captures some of the change drivers in Table 2.2 (e.g., change drivers related to the organizational context). Other change drivers are not considered.

Drawing upon the theoretical insights provided by these two models, we propose a framework to model how various social and technical change drivers, identified in Table 2.2, are connected to each other to develop the process of change in IS requirements.

Research Methodology

Our research methodology is based on the *design science research* in IS literature (Hevner et al. 2004; Peffer et al. 2007). We propose two information technology artifacts, a model and a method. These two artifacts are considered as solutions to extend the capabilities of IS analysts and designers in understanding the process of requirements change and anticipating the potential future change requests.

According to Woo et al. (2014), an artifact in design science research can be a "technological design" that is developed based on the previous knowledge and theories, but it enables researchers to observe systems from a new angle that was not previously possible. The STS approaches mentioned in the previous section, particularly the punctuated socio-technical change model of IS and socio-technical requirements elicitation process model, inspire us to observe the change drivers and their interconnection from a new angle. However, they do not cover many of the change drivers in the existing requirements studies.

In the following section, we introduce the **Socio-Technical Change Framework** as a model and **Socio-Technical Requirements Change Method**. These two artifacts can be used along with any requirements determination technique to enable researchers to observe the change drivers (in Table 2.2) and their interconnections from a new angle that was not previously possible.

The Socio-Technical Change Framework

Our Socio-Technical Change Framework elaborates on requirements change processes through four levels: ***environmental context***, ***organizational context***, ***IS development context***, and ***IS requirements context***. The first three levels are drawn from Table 2.2, and they show that requirements change processes can be originally initiated from the environment, organization, or IS development project. We add the fourth level to show the impact of the change drivers in any of the first three levels on IS requirements. These four levels correspond to those ones in the punctuated socio-technical change model of IS (Lyytinen and Newman 2008). However, the main differences are: 1) we expand the work system level, based on the requirements change literature, to include the organizational context; and 2) we add the IS requirements level.

Each of the environmental context, organizational context, and IS development context is composed of a socio-technical system. In accordance with the punctuated socio-technical change model of IS (Lyytinen and Newman 2008), changes in any of the social or technical elements of these systems can generate *two types of responses* in the framework. In the *first type of response*, these changes lead to further imbalances within the same system/level. As a result, the misaligned elements should be adjusted to keep the whole system balanced or stable. In the *second type of response*, these changes traverse across and develop further changes in other levels. Figure 2.2 demonstrates the content of each level and shows some of the interdependencies between the four levels of the Socio-Technical Change Framework.

In Table 2.3, we associate the various change drivers collected from the literature (previously summarized in Table 2.2) with the social and technical components within the environmental, organizational, and IS development levels. This information will be used to explain our proposed framework below. The change drivers in the right column are considered as some examples to display the social and technical components in each layer. In practice, there can be other social and technical change drivers that have not been investigated by the literature so far (e.g., change drivers related to processes in the environmental context). Our framework still is able to show those change drivers, their interconnections, and their impact on the requirements changes.

Table 2.3 Socio-Technical Elements in the Socio-Technical Change Framework

Level	**Socio-Technical Components in Each Level**	**Change Drivers**
Environmental context	Structure	Market demand and attitude towards a technology, regulations and laws, economics
	Actors	Competition
	Technology	Available related systems, available technologies
Organizational context	Structure	Business goals and strategies, policies, values, power and politics, management style and control systems, rules
	Actors	Job description and roles, technical team capacities
	Tasks	Business processes
	Technology	Technology policies, business value of technology, technology-enabled work practices
IS development context	Structure	Scope and goals, complexity level of the problem under analysis, management and culture, project cost and time, delivery strategies
	Actors	Participants with their expectations and experience, skills of analysts and developers
	Tasks	Development and programming activities and their challenges
	Technology	Requirements elicitation technique, technology types and capabilities

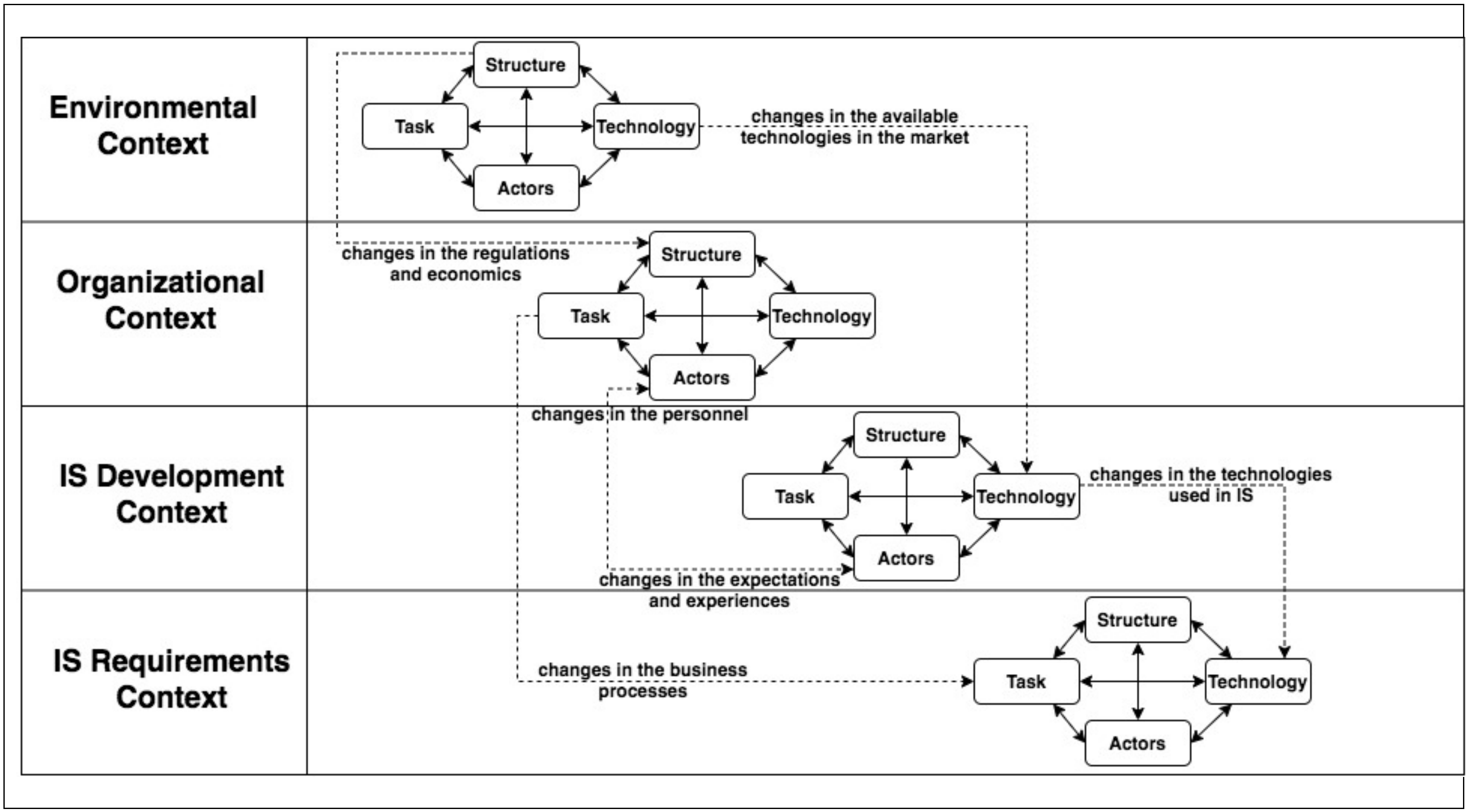
We note that the categorization of change drivers into one of the four social and technical elements — actors, structure, task, and technology — depends on our interpretation of the context. This is mentioned by Lyytinen and Newman (2008) as well. For example, we can categorize "competition" as either an actor or a task dimension of the environment. The exact categorization does not matter, as long as we analyze the impact of the change in the competition on other social and technical components in the same level (environmental context), as well as its effect on the other levels (organizational and IS development contexts). These two types of response to a change are elaborated in more details below.

Various requirements change processes can be commenced from the environmental context. The first group of change drivers in Table 2.3 demonstrates a number of sources for these processes. For example, if the "government regulations and laws" are changed, this impacts the "available technologies on the market". In other words, the technology component within the environmental context adapts to the original change driver. Also, it can generate additional changes in the "organizational strategies and rules". In other words, the structure component in the organizational context evolves in accordance with the original change driver. Changes in the environmental context can also pass through the IS development context. For example, changes in the "available technologies on the market" can change the "technologies that are used in a particular project". In other words, the technology component in the IS development context adapts to the original change driver. All of these changes finally will arrive at the IS requirements level and create further changes in the requirements (e.g., by adding new requirements or modifying/removing the existing ones).

Similarly, changes might be initiated from the organizational context or IS development context. Changes from the organizational level can directly or indirectly, through the IS development level, make future changes in IS requirements. For instance, if some "personnel" leave the organization or "new managers" come in, the "participants involved in the IS project" are also changed. Consequently, new stakeholders will propose new expectations and requirements.

Figure 2.2 demonstrates a few examples of the change processes initiated from the environmental context, organizational context, and IS development context in the framework. It should be emphasized that the interdependencies between these levels are not limited to the links shown in Figure 2.2. Each of the change drivers in Table 2.3 can potentially initiate a chain of change within the same level or across other levels.

Figure 2.2 The Socio-Technical Change Framework (Illustration of a Few Examples)



IS Requirements Change

The last level in the Socio-Technical Change Framework is the IS requirements context. Each requirement piece is associated with a socio-technical composition, which explicitly or implicitly brings information about the:

1) **Actors**: People who own the requirement and have specific skills to use the requirement to achieve their business goals;

2) **Task**: Specific tasks that actors should perform along with using the technology (e.g., their organizational duties and responsibilities);

3) **Structure**: The business justification or organizational regulation behind the requirement; and

4) **Technology**: Particular technical capabilities and features that the requirement is based on.

Any change driver, from the first three levels, can generate one or more of the following responses at the IS requirements level: creating new requirements, modifying the existing requirements, and removing the existing requirements.

Regarding the modification situation, each change driver can impact any of the four social and technical elements of a requirement. For instance, if the “business processes” in an organization are modified — changes in the task component of the organizational context — they will adjust the task element of the related requirements (e.g., requirements of people who participate in those particular business processes). Or if the IS project participants decide to change some aspects of the “technical capabilities and features” — changes in the technology component of the IS development context — then the technology element of the related requirements will adapt to this change as well.

In general, anytime a social or technical component associated with a requirement piece is changed, this can lead to *two types of further changes* at the IS requirements level. In the *first type*, other components within the same socio-technical composition should be adjusted to keep the whole socio-technical composition stable. For example, if we decide to change the “technology” component of a requirement, we need to verify that the “actors” possess the required skills to work with the new technology; otherwise there will be a misalignment issue between the actors and technology components. If we cannot address the misalignment issue here, we will face its consequence in the future. In this example, the users will not be able to successfully apply the IS to perform their business responsibilities at work, and they will come up with their change requests later.

In the *second type of change*, the revised requirement will influence other related requirements. When we study changes in requirements, we should remember that many requirements in a project are dependent on each other (Jayatilleke et al. 2018; Zhang et al. 2014; Dahlstedt and Persson 2003). For example, sometimes the achievement of one requirement depends on the achievement of another requirement, or the achievement of one requirement conflicts with the achievement of another requirement (Dahlstedt and Persson 2003). In the Socio-Technical Change Framework, we recommend that the socio-technical components of each requirement are used to find the dependent requirements. More particularly, when a requirement of an actor is changed, we should search related requirements in task, structure, and technology that are associated with this actor. For example, if two requirements of an actor are about the same task or two dependent tasks, the change in one requirement most probably influences the other requirement. We also should search the related requirements of other actors who are connected to the focal actor through various organizational links and hierarchies. This is further elaborated in the next section.

As mentioned in this section, the Socio-Technical Change Framework proposes a novel perspective to observe change processes in IS requirements. To the best of our knowledge, there is no similar model in the existing requirements literature that illustrates the requirements change process from the emergence of a change driver to the final requirements change. This framework shows how different change drivers identified by the literature can be interconnected with each other, and how they jointly can create a chain of change in IS requirements. This elaboration helps analysts gain a deeper understanding of the reasons and processes behind the change requests of their clients. In addition, this framework provides analysts with a solution to validate the requirements and anticipate some of the future changes in them based on the aforementioned relationships. The Socio-Technical Requirements Change Method illustrates these two benefits of the framework in the following section.

The Socio-Technical Requirements Change Method

Drawn upon the proposed framework, the Socio-Technical Requirements Change Method introduces a new solution for managing IS requirements, specifically for evaluating the collected requirements and anticipating the potential changes in them. This method focuses on the change mechanisms within the IS requirements level, which were discussed in the previous section.

Evaluating the existing requirements

After collecting the requirements of users, IS analysts need to confirm the accuracy and consistency of these requirements. As explained by the Socio-Technical Change Framework, the socio-technical composition associated with each requirement piece should remain in a stable state, with no misalignment between the components. Otherwise, some social or technical components might need to change to keep the whole system stable — *this is a change in the requirement*. If the misalignments are not detected and resolved at the beginning of an IS development project, the inaccuracy issue remains hidden until it shows up in further tests or after the implementation. We recommend that analysts assess the alignment within each requirement piece and revise the requirement if needed as soon as possible — *tackling the change at source*.

The Socio-Technical Requirements Change Method proposes a solution to perform the assessment and revision. Table 2.4 demonstrates guidelines to detect, resolve, and take advantage of the misalignments within a socio-technical composition associated with each requirement. The majority of these guidelines originally draw upon the previous studies (Lyytinen and Newman 2008; Sarnikar et al. 2014), and we operationalize them in the context of requirements change.

According to the Socio-Technical Change Framework, anytime there is an adjustment in a requirement piece, other dependent requirements might be influenced as well. Therefore, in addition to resolving the misalignment within each requirement piece, it is important to track the consequences of the requirements change on other dependent requirements. If those requirements could be tackled and revised as early as possible, IS analysts would prevent potential future inconsistencies and changes in the IS.

Table 2.4 Detecting and Resolving Misalignments within a Requirement

Socio-Technical Alignment	Assessing the Alignment	Resolving or Taking Advantage of the Misalignment
Actors – Technology	Do the actors possess the required skills and knowledge to work with the proposed technology features? If not, there is a misalignment between the actor and technology components.	Training and educational programs should be conducted to improve the knowledge and skills of actors. Also, the technology features should be revised to better fit the knowledge and skills of actors.
Actors – Task	Are the tasks (or business activities) associated with the requirement originally assigned to these actors in the organization? Also, are these actors capable of performing the tasks? If not, there is a misalignment between the actor and task components. If there is any misinformation about the business domain, it should be tackled here.	This is an organizational problem that should be communicated with the clients. They may need to adjust their work system or job descriptions. Also, they may request new requirements to address their business needs (e.g., using technology features to fill the gap).
Actors – Structure	Are the actors' skills and knowledge aligned with the expectations of the organization? Are their skills and knowledge sufficient to fulfill the decisions of the IS project? If not, there is a misalignment between the actor and structure components. If there is any misinformation about the business domain, it should be tackled here.	This is an organizational problem that should be communicated with the clients. They may need to adjust their work system or provide the IS users with additional training.
Technology – Task	Are the proper technology features chosen to support the business tasks? If not, there is a misalignment between the technology and task components.	The technology features should be revised to better fit the business tasks.
Technology – Structure	Are the selected technology features aligned with the strategies, expectations, and values of the organization? Can they fulfill the decisions of the IS project? If not, there is a misalignment	The technology features should be revised to better fit the expectations of the organization and IS project.

Socio-Technical Alignment	Assessing the Alignment	Resolving or Taking Advantage of the Misalignment
	between the technology and structure components.	
Task – Structure	Are the tasks aligned with the expectations and values of the organization? Can the tasks fulfill the decisions of the IS project? If not, there is a misalignment between the task and structure components. If there is any misinformation about the business domain, it should be tackled here.	This is an organizational problem that should be communicated with the clients. They may need to adjust their work system to address the misalignment.

To find the impacted requirements, the Socio-Technical Requirements Change Method proposes that IS analysts investigate other dependent requirements owned by the same actor as well as the dependent requirements owned by the *role set* of that actor. For any organizational role, a role set is any particular group of the organizational roles that depend on the focal role to accomplish their activities in the organization (Kahn et al., 1964). Most often, the role set members for any given role include its supervisors, subordinates, and co-workers. The dependent requirements need to be adjusted according to the change in the original requirement.

The following summarizes the main steps in the Socio-Technical Requirements Change Method to evaluate the existing requirements:

1. Choose a requirement and assess the alignment within the socio-technical composition associated with the requirement.
2. Resolve the misalignment, if any.
3. Analyze the impact of the change in this requirement on other dependent requirements of the same actor. For each dependent requirement:
   3.1. Check the consistency between the original requirement and this one.
   3.2. If any inconsistency exists:
      3.2.1. Adjust this requirement.
      3.2.2. For this requirement, perform the whole process (from step 1).

4. Analyze the impact of the change on other dependent requirements of the role set of the focal actor. For each dependent requirement:
   4.1. Check the consistency between the original requirement and this one.
   4.2. If any inconsistency exists:
      4.2.1. Adjust this requirement.
      4.2.2. For this requirement, perform the whole process (from step 1).

Anticipating a future requirements change

The Socio-Technical Change Framework elaborates on the requirements change processes, from the emergence of the change drivers to the final changes in IS requirements. Anytime a change driver emerges, IS analysts can use the change processes discussed in the Socio-Technical Change Framework to anticipate further changes in requirements before they happen.

The Socio-Technical Requirements Change Method translates the theoretical change processes in the framework to the practical guidelines. Specifically, the Socio-Technical Requirements Change Method proposes that, when a change driver emerges (in the environmental context, organizational context, or IS development context), IS analysts anticipate the consequence of the change driver on IS requirements. It is expected that the social or technical components of some requirements adapt to the change driver, which results in the potential misalignment within each of those requirements. By employing the guidelines in Table 2.4, IS analysts can detect the potential misalignments and resolve them. If the misalignment cannot be tackled as early as possible, they will generate further problems and change requests in the future. By detecting and resolving the hidden misalignments within each requirement, IS analysts can anticipate and prevent some future changes before they happen. Consequently, this will mitigate the cost of later corrections.

As mentioned earlier, when a requirement changes, it may impact its dependent requirements as well. The Socio-Technical Requirements Change Method proposes that IS analysts examine the dependent requirements, owned by both the actor and its role set members. If there is any inconsistency between the changed requirement and its dependent requirements, they should also be adjusted accordingly. Otherwise, the inconsistencies generate further problems and change requests in the future. By detecting and resolving the inconsistencies between requirements, IS analysts can anticipate and prevent some future changes before they happen.

The following summarizes the main steps in the Socio-Technical Requirements Change Method to anticipate the future requirements change:

1. Anytime a change driver emerges, identify the IS requirements which are potentially impacted by the change driver. For each of these requirements:
2. Apply the anticipated change.
3. Assess the alignment within the socio-technical composition associated with the requirement. Resolve the misalignment, if any.
4. Analyze the impact of the change in this requirement on other dependent requirements of the same actor. For each dependent requirement:
    4.1. Check the consistency between the original requirement and this one.
    4.2. If any inconsistency exists:
        4.2.1. Adjust this requirement.
        4.2.2. For this requirement, perform the whole process (from step 1).
5. Analyze the impact of the change on other dependent requirements of the role set of the focal actor. For each dependent requirement:
    5.1. Check the consistency between the original requirement and this one.
    5.2. If any inconsistency exists:
        5.2.1. Adjust this requirement.
        5.2.2. For this requirement, perform the whole process (from step 1).

Discussion of Evaluation

Based on the change mechanisms discussed in the proposed framework, the Socio-Technical Requirements Change Method presents a solution to evaluate the existing requirements and anticipate the potential requirements changes. This method is used to demonstrate the use of the Socio-Technical Change Framework in modeling and explaining various change processes in IS requirements. In the design science research, demonstrating the use of an artifact is the first step in the process of evaluation (Peffer et al. 2007).

Further steps should be taken to assess the utility of the Socio-Technical Requirements Change Method and consequently measure how well the Socio-Technical Change

Framework explains the change processes in IS requirements. While the foundation of the Socio-Technical Change Framework is based on previous rigorous studies, the evaluation of this framework is essential, particularly in a real-world setting. We have attempted to evaluate the framework but encountered the following challenges.

First, we attempted to assess how successful the Socio-Technical Requirements Change Method is to anticipate the requirements changes that emerge in a project. To collect data, we contacted two consulting firms. They accepted to confidentially share the requirements documents for two projects with us, one off-the-shelf system project and one mobile application project. Unfortunately, not all the required data was documented (e.g., the details about the change drivers and requirements changes over time). In general, we found it rare that requirements changes are clearly documented.

Second, we interviewed two business analysts. We asked them to match their previous experience in requirements change to the processes modeled by the Socio-Technical Change Framework. We inferred that business analysts generally possess a macro-level perspective of the change processes. More specifically, it seemed to be difficult for our interviewees to translate those abstract processes into the specific interactions between the social and technical components. Therefore, we did not continue inviting more participants to the interview.[5]

For future research, we suggest that case studies should be conducted to evaluate the Socio-Technical Change Framework and Socio-Technical Requirements Change Method in ongoing IS development projects. Specifically, we need to educate the analysts on these tools and ask them to use the tools along with their requirements elicitation techniques. This enables us to measure the usability, functionality, and efficiency of these two artifacts and identify any potential complication in their application. We can then move to measure the utility, benefit, and effectiveness achieved by employing these artifacts (Venable et al. 2016).

The challenge here is to find such a project. It is rare that information systems development projects are not constrained by time, budget, and willingness of analysts to try an unproven framework and/or method. This is why such an evaluation would form a separate study, which is out of the scope of this essay and we leave it for future research.

[5] Originally, we had planned to continue these interviews and develop our second study based on them. However, due to the difficulties mentioned above, we later decided to change the research question and design in our second study.

Conclusion

Research Summary

In this chapter, we collected the reasons for change in IS requirements from the literature and synthesize them. The extant requirements studies have identified a wide spectrum of change drivers ranging from the social factors to technical ones. Yet, we could not acquire further information on how various social and technical change drivers are interconnected to each other and how they can jointly develop the changes in IS requirements.

By employing a design science research methodology, we proposed the Socio-Technical Change Framework. This framework models and illustrates various requirements change processes, which start from the emergence of a change driver and end at the final changes in IS requirements. We then introduced the Socio-Technical Requirements Change Method, as a demonstration of the use of the proposed framework. This method provides IS analysts with a set of guidelines to evaluate the existing IS requirements and anticipate further changes in them.

Contributions

This research makes a number of contributions to both research and practice. First, this study contributes to the existing body of knowledge on IS requirements change. We propose the Socio-Technical Change Framework, as a technical design artifact. This framework integrates all the change drivers, which have been identified by the previous studies. This framework elaborates on how these change drivers jointly create the changes in IS requirements. More specifically, the proposed framework demonstrates that requirements change is the outcome of several interactions among various social and technical elements — actors, structures, task, and technology — across the environment, organizations, and IS development projects. We cannot find any similar model in the existing literature that explains how social and technical change drivers are interdependent on each other to create the changes in IS requirements.

Second, the results of this study also have practical implications. The Socio-Technical Requirements Change Method can be used as a requirements analysis tool by IS analysts.

This method supports analysts in evaluating the requirements they collect from their clients. Also, the aforementioned framework provides IS analysts with a deep understanding of the logic behind requirements change. By employing these two artifacts, analysts can anticipate the consequence of the emergent changes in the environment, organization, or IS development project on requirements. As a result, analysts can anticipate some of the potential changes in IS requirements and modify the requirements if needed in advance. This can significantly mitigate the risks and costs associated with late corrections and change requests. Even if IS analysts, designers, or project managers decide to not apply the anticipated changes instantly, at least they are prepared for the change scenarios in the future.

Finally, the Socio-Technical Change Framework and Socio-Technical Requirements Change Method can be used in any requirements determination project. They can explain and anticipate the requirements changes independently of the type of domain (e.g., health care, finance, marketing, etc.), IS development methodology (e.g., waterfall and agile), and requirements elicitation technique (e.g., goal-oriented, agent-based, scenario-based, etc.).

Chapter Three

Meeting Requirements Change in Contemporary Information Systems Projects

Introduction

Understanding IS requirements change has been a popular topic for many of the requirements studies over the past four decades. During this time, information technologies have grown exponentially and brought innovative digital affordances, capabilities, and features more or less every day. Similar to the individual users of these technologies, organizations have also significantly adapted to new technology systems. In today's world, many firms cannot even survive without digital technologies, including but not limited to enterprise resource planning systems, e-commerce platforms, mobile systems, social media platforms, cloud technologies, business analytics tools, and so on. Technological improvement and innovation over the past decades have not been limited to the ***type*** of technologies used by organizations. For many firms, their ***IS development practices*** have been also shifted from ordering custom-built solutions towards purchasing packaged products. Instead of designing and building a system from scratch, these firms prefer to purchase on-premise systems or subscribe to cloud-based systems that have been previously tested and trusted. In April 2019, Gartner estimated the future growth of the cloud services industry would be about the three times of the growth of the overall IT industry through 2022 (a total of 331.2 billion US dollars revenue).[6]

By considering all of these various options in technologies nowadays, it is important and interesting to study the change processes in requirements in today's IS projects. Due to the technology evolutions mentioned above, we suspect that requirements changes and their root causes are different now. For example, many requirements in off-the-shelf systems

---

[6] Source: https://www.gartner.com/en/newsroom/press-releases/2019-04-02-gartner-forecasts-worldwide-public-cloud-revenue-to-g

(OTSS) are designed already and built into the packaged solutions, but still evidence shows that these requirements evolve over time (e.g., because of unstructured business processes or technology capacities in a firm) (Schneider et al. 2018). Also, modern technologies, such as artificial intelligence and data analytics, may not be well understood by many clients yet, which implies a more complex requirements determination process. Therefore, it is essential to gain a deeper insight into what reasons result in change in IS requirements when dealing with contemporary projects and investigate how recent technologies and systems development approaches impact requirements change.

Among the studies we reviewed in Table 2.2, only a few of them discuss the effect of contemporary technologies or system development solutions on requirements and their change. For example, Schneider et al. (2018) studies the impacts of cloud-based solutions on post-implementation changes through a case study. In a theoretical piece, Jarke et al. (2011) underlines the role of new trends, including OTSS, e-commerce platforms, and media applications, in managing requirements. Tuunanen and Kuo (2014) depictes the influence of culture on requirements in mobile systems. However, this study does not explicitly elaborate on what part of the findings can be different if the study is conducted through other technologies rather than mobile platforms. In general, it seems that existing requirements literature provides little explanation on the specific influence of recent technologies and development solutions on requirements changes.

In order to address this gap, this research answers the following research questions: ***What are the root causes of requirements change in contemporary IS development projects? How do contemporary technologies and IS development practices impact the processes of requirements change?*** Essentially, we concentrate on requirements changes in OTSS, including both cloud-based and on-premise solutions, as well as contemporary technologies, including mobile applications, e-commerce platforms, algorithm-based solutions, machine learning tools, and data analytics technologies. We approached this study from a qualitative research methodology and collected information and evidences from the real-world experiences.

Research Design

To tackle our research questions, we sought to explore requirements changes that IS consultants meet in their systems development projects. According to Miles and Huberman (1994), qualitative research is a powerful approach to describe things, explain how things are connected together, and answer why things happen in real life. One of the main tasks in qualitative research is to *"explicate the ways people in particular settings come to understand, account for, take actions, and otherwise manage their day-to-day situations"* (Miles and Huberman 1994, p. 7). Therefore, we decided to use qualitative research methodology to collect data on what requirements changes happen for IS consultants when dealing with contemporary technologies and systems development methods; why those changes occur; how IS consultants observe the impact of modern technologies and solutions on IS requirements change; and finally how they manage the change requests. Using an inductive approach helps us explore the aspects of requirements change in modern IS projects that have not been demonstrated yet theoretically or empirically.

There are various methods in qualitative research, including interviewing, field observation, and document analysis. We chose interviewing to answer our research questions. Interviewing is the most widely used method in qualitative research (Schultze and Avital 2011; Polkinghorne 2005), and it is a proper choice when a direct observation of individuals' experience is not accessible to the researcher. In these situations, *"interviewing seeks to engage subjects directly in a conversation with the researcher so as to get a first-person account of the participant's social reality"* (Schultze and Avital 2011, P. 3) We conducted a set of interviews with IS consultants to collect data about their experience of requirements changes.

As will be explained in the following sections, the data obtained from in-person interviews provided us with evidence about the impact of modern technologies and IS development solutions. To verify presumptive results and increase the validity of our findings (Miles and Huberman 1994; Gubrium et al. 2012, Chapter 6), we collected additional data by conducting a set of qualitative surveys. The remainder of this chapter explains the details of our data collections and results.

Interviews

Data Collection

We interviewed 10 IS consultants, including nine business analysts and one project manager from seven different firms, who had sufficient experience in the area of changing requirements in contemporary IS development projects.

We followed two strategies to identify our participants. In the first strategy, we contacted the chief information officers or a project manager in three companies, one from the financial and banking sector and two consulting firms, and requested that they share our invitation letter with their employees (see Appendix A and Appendix B). From each of these companies, two IS consultants, mainly senior business analysts, whose professional background matched with the criteria in the invitation letter volunteered to participate in our study. In the second strategy, we directly contacted four IS consultants, whom we previously knew from their association with the University of British Columbia, and invited them to our study (see Appendix B), one from a telecommunication firm and three from three different consulting firms. Their backgrounds also matched with the criteria in the invitation letter, and they volunteered to participate in our study. In total, our 10 interviewees had experience of requirements changes in various domains, including but not limited to health care, higher education, telecommunication, financial and banking, insurance, marketing, retail, wholesale, government, and aerospace. Table 3.1 illustrates the background information of our interviewees.

Semi-structured face-to-face interviews were conducted to collect data for this phase of the research. Only one of them was a phone interview. We sent out the questions (see Appendix C) and consent form to our interviewees in advance, so they would have a clearer understanding of the interview process and questions. We had eight standard questions and made changes in them or asked additional questions depending on participants' answer to the initial questions. Interviews generally lasted between 30 and 75 minutes, and the main part of the interview focused on learning about the requirements change processes that our participants had experienced in their projects. All the interviews were audio recorded and transcribed. We first used a professional auto-transcription tool. Later, all the initial transcripts were checked and confirmed by the author of this dissertation. Also, for a few of the transcriptions, a research assistant helped us check the accuracy of software-generated transcripts before the final confirmation.

Table 3.1 Background Information of Interview Participants

Participant ID	Work Experience	Company/ Project Type	Contemporary Projects Features
Participant #1	15 years, Senior business analyst	Consulting firm	OTSS (cloud-based and on-premise solutions), etc.
Participant #2	9 years, Senior business analyst and Solution architect	Consulting firm	OTSS (cloud-based solutions), E-commerce, etc.
Participant #3	7 years, Senior business analyst	Consulting firm	Mobile systems, E-commerce, etc.
Participant #4	24 years, Senior consultant in strategy, transformation, and change management	Consulting firm	OTSS (Cloud-based solutions), E-commerce, etc.
Participant #5	4 years, Senior business analyst	In-house consulting team	Robotic process automation
Participant #6	40 years, Senior business analyst	Both consulting firm and in-house consulting team	OTSS, E-commerce, Machine learning, Data analytics, etc.
Participant #7	10 years, Business analyst	In-house consulting team	OTSS, Big data
Participant #8	4 years, Business analyst and Business consultant	In-house consulting team	OTSS (cloud-based and on-premise solutions), Mobile systems, E-commerce, Virtual teams, Algorithm-based systems, Big data, Machine learning
Participant #9	1.5 years, Business analyst	In-house consulting team	OTSS
Participant #10	2.5 years, Senior business analyst	Consulting firm	OTSS (cloud-based solutions), Mobile Systems, Virtual teams, Big data, Data analytics, Machine learning

Data Analysis

Our data analysis generally followed the guidelines provided by Marshall and Rossman (2016). The author of this dissertation iteratively read transcripts, coded the data, generated categories of change drivers from the codes, and refined the categories. She met with her supervisor after interviewing every one or two new participants to discuss the categories and their intended meaning. This resulted in further refining the categories and clarifying their meaning.

We used *NVivo* software program for our data analysis. In order to code each transcript, we thoroughly read the text. Anytime we found new information about change drivers, role of contemporary technologies, role of OTSS, or role of modern world, we assigned a new code to that piece of information. If information in the text confirmed an existing code, we then assigned the existing code to that piece of information. Each information chunk could be associated with as many codes as required.

Besides the main information about change drivers and contemporary context, we coded other contextual information about IS development methodology approaches (e.g., agile and waterfall), size of companies, users' behavior, change management strategies, time of change requests (e.g., during development and after implementation), etc. This contextual information helped us better understand details of change processes.

Drawing upon the codes in each transcript, we generated categories of change drivers, characteristics of contemporary technologies, characteristics of OTSS, and characteristics of modern world. We tested our categories against new data. After coding a new transcript, we compared the new codes with the existing categories. If we observed the new codes were revealing a new category (e.g., a new change driver), we then would create a new category based on them. If the new codes provided additional insights into an existing category, then we would add them to that existing category and refine it. For example, the impact of *new people* and *new business process* resembled the impact of other organizational changes on IS requirements, thus we added them as new dimensions of *changes in organizational factors*.

The emergent codes also guided us to modify some questions in our next interviews. For example, after we had inferred the significant effect of the *continuous exposure of IS users to fast-changing technologies* on requirements changes, we explicitly asked about the opinion of IS consultants about this effect in our next interviews.

It should be noted that we did not directly use the change drivers from the previous study as our initial codes (or theory-generated codes). However, we think they implicitly influenced our analysis and judgment in coding and categorizing the change drivers here. This happens in qualitative research, for example when the researcher tries out the codes from the literature (Marshall and Rossman 2016).

In total, 14 categories of change drivers were identified representing the main reasons of change in IS requirements in contemporary IS development projects. Table 3.2 summarizes these 14 categories. As shown in this table, Participant #7, Participant #9, and Participant #10 did not identify any new change driver, but they confirmed our previous findings. This brought us to a point of data saturation, so we did not continue inviting new participants. To increase the accuracy and validity of the findings, later we checked the completeness of the change drivers with our survey participants. This will be discussed in detail later. Survey results confirmed the change drivers identified from interviews. They did not show any new change driver category. Survey findings seemed to be sufficient, and therefore we did not design a second coding phase.

Findings

All of our interviewees believed that change in requirements is an inevitable part of their projects.

> *"I can tell you, yes it always happens." (Participant #3)*
>
> *"Change requests are pretty much within my vocabulary. I use, I don't know, how many times a day." (Participant #1)*
>
> *"I have never worked on any system in my life where the clients have not asked for changes during development or after implementation." (Participant #6)*

The 14 change drivers will be elaborated in the remainder of this section. It is important to note that, although contemporary IS development practices (e.g., OTSS) and modern technologies (e.g., mobile systems and data analytics) formed the main frames in our interviews, we inferred that requirements changes are not directly drawn from the features and attributes of modern solutions. These requirements changes are still rooted in various social and technical aspects that have been around since the emergence of classic IS projects, a few decades ago. However, the modern solutions, and our contemporary world in general, play novel, interesting, and important roles in relation to those classic causal relationships.

Table 3.2 Change Drivers of IS Requirements Drawn upon the Results of Interviews

Change Driver ID	Change Driver Name	Informants
#1	Lack of involvement of correct clients	Participant #1, Participant #2, Participant #3, Participant #9
#2	Oversight of details and recalling issue	Participant #1, Participant #3, Participant #6, Participant #7, Participant #9, Participant #10
#3	Inability to articulate the requirements	Participant #6
#4	Lack of transparency from business analysts	Participant #1, Participant #8
#5	Lack of a deep understanding of the project	Participant #8
#6	Business Process ambiguity in organizations	Participant #1, Participant #2, Participant #10
#7	Conflicts between business needs and technical standards	Participant #5
#8	Change in environmental factors	Participant #1, Participant #2, Participant #3, Participant #5, Participant #6, Participant #7, Participant #8, Participant #10
#9	Changes in organizational factors	Participant #1, Participant #2, Participant #5, Participant #6, Participant #10
#10	Changes in the IS project	Participant #1, Participant #6, Participant #7, Participant #10
#11	Improving user experience	Participant #2, Participant #4, Participant #7
#12	Changes in user expectations	Participant #1, Participant #2, Participant #4, Participant #7, Participant #8, Participant #10
#13	Changes in the technical knowledge of users	Participant #8
#14	Changes in interconnected systems	Participant #3, Participant #5, Participant #7, Participant #9

Now, we discuss the 14 change drivers:

1- Lack of involvement of correct clients

When the actual end-users and subject matter experts do not participate in the process of requirements gathering, business analysts cannot successfully collect all business requirements of the people who will work with the intended IS in the future. As a result, the IS will be designed based on incomplete requirements, and they will not be able to fulfill the whole needs of the organization. Consequently, clients will face further difficulties in the future and come up with change requests. For instance, in some cases, the management decides on behalf of their employees or chooses some wrong representatives, who do not have the complete information about the daily processes of end-users (first subcategory).

> *"So what happens is that upper management says this is what we want. We want to upgrade this, or we want to have something new. They won't necessarily think it through, and the biggest thing is they don't speak to the end-user. The only thing is from their perspective, this is how I want to collect data and this is how I want to be able to drive my sales, drive my service performance, and so forth […] Whereas end-users are more about day-to-day, how do they function in a day. And then there is always a gap, that's where a lot of the change management comes into the play, or the change requests. Because you'll develop something but then you realize that actually management or the sponsors have not talked to the actual end-users […] Sometimes they send someone who doesn't care per se, or may not be fully aware of what's being expected from them." (Participant #1)*

> *"The right people and experts, they were not as involved as they have to be from the beginning. […] They don't know when to engage them and which folks to engage at the right time." (Participant #9)*

Lack of internal communication and disagreement between clients are also reasons for the change in requirements (second subcategory). Sometimes, clients change their requirements continuously, especially when there are various stakeholders in a project.

> *"They want something, you put it in front of them. And then another person comes in. They want something else, and they do not agree. There is always that back and forth.*

We usually say two reviews and then we have to lock it. Because you can't wait for a very [long time to get the final approval]. Different stakeholders, it's very hard to come to consensus between them. A lot of times, those changes come from the fact that we are working with one department, and we get agreement on everything. And then once we are done, the boss above them has to agree and they say that oh no there's something else." (Participant #3)

Impact of OTSS: From the interviews, we discovered that using OTSS can mitigate the impact of this change driver. This is because the majority of technical features are already built into packaged solutions, and it is more convenient for non-technical users to get involved in IS development projects.

2- Oversight of details and recalling issue

If clients and analysts forget to consider some aspects of IS requirements at the beginning of a project, the intended IS will not be designed based on the complete list of requirements. This will result in future problems, and clients will come back with change requests to resolve their issues.

"So sometimes not all the information is given at the beginning, people just tend to not remember something or not define the proper need for that requirement. Some stakeholders may not even read the document that you produce, and then later in a day they will come back and say hey what about this? And then we have to say well if you read the document, which you approved, you would have known that it's not there. So yeah, so sometimes they forget or you know, they don't really divulge all the information that they have." (Participant #7)

"When you are doing requirements gathering, you have to make sure you cover the things going well but you also have to deal with what happens when you go off the tracks and that's one of the major causes for change requests. […] An edge case means boundary. And so quite often during the course of discovery, you never cover all the edge cases. […] So edge cases are a major cause for changing, because you have to think about every possible thing that could happen in the course of a user interacting with your solution." (Participant #6)

"Sometimes things get lost. I mean sometimes clients see something in the beginning, they don't mention anything at all. And towards completing the training, they're like, hey by the way we need this. I am like what? You didn't mention it. Maybe we should have asked, maybe it's our fault. Who knows? So it happens." (Participant #10)

Impact of the modern world: According to the interview results, there are many global and multinational projects in our contemporary world. As the scale of a project increases, there is a higher possibility of missing information in the process of requirements collection.

"They just didn't consider, you know, all these little things that really come into play. Like there's no one person who's going to know it." (Participant #1)

The interview results show that the expertise of analysts is very useful in controlling this change driver. For example, the more experience gained in a particular domain, the greater the attention of analysts is to details and specific edge cases within that domain.

"Finance has certain edge cases, health care has edge cases, and insurance has edge cases. Certain businesses within certain industries will have different edge cases. Is it a multinational or the American only company? Is that a North American only company? You know what language do they speak? Is it a multilingual company? Do they only do business in English or English and French? And so the more experience of team has and dealing with all of these variations allows them to identify edge cases because they've seen them before." (Participant #6)

Impact of OTSS: From the interviews data, we interpreted that packaged solutions can significantly prevent the missed information in a project. This is because vendors continuously improve their OTSS based on new experiences and cases.

3- Inability to articulate the requirements

Sometimes, clients may not be able to precisely describe their requirements. For example, clients may use words and terms that cannot accurately convey their intended ideas. In these situations, analysts collect requirements based on what they hear. As a result, they design IS based on inaccurate requirements. These IS will not completely fulfill the needs of clients, so clients will come back with change requests later.

> *"So part of the change cycle is that the client will say I know what I said, but that's not what I meant. They will say something but they meant something else. […] You said, we wrote it down. You signed off on it. Now you're coming back and say that's not what you meant. We built what you told us to build. Now you're arguing that we did not build what you told us because you meant something else when you said that." (Participant #6)*

The interview results suggested that documenting all the details and making a glossary of all the specific terms used in requirements are essential steps for good communications between clients and analysts.

> *"Clients always believe it's a defect you got it wrong, and we always believed that it's a change that we got it right. And this is why it is so important that during the requirements gathering phase you document absolutely everything to ensure that there is no miscommunication and no misunderstanding. That you have a glossary so that you can say this word means this, and you don't use the same word for two different concepts." (Participant #6)*

Impact of modern technologies: The findings explained that some modern technologies, such as machine learning and data analytics, are very complex to understand by non-technical users. This results in more challenges for non-technical users to articulate their requirements.

Impact of OTSS: The results revealed that OTSS significantly support users to visualize the future systems at the beginning of projects. Consequently, users are able to better elaborate on the requirements they have in their mind.

4- Lack of transparency from business analysts

When business analysts explain the solutions and their consequences with higher transparency, clients can more clearly understand the alternative solutions and outcomes. Therefore, clients will face less surprise in the future and have fewer change requests.

"It's just a matter of setting the expectations. So making it very clear and transparent to customers, to really understand what it is that they will be getting. You know, I find transparency being the most helpful for me." (Participant #1)

"[Analysts] have the experience. They should tell the user what would be the impact or what is the alternative, so they can have a look and feel of it, before going to the next level." (Participant #8)

Impact of modern technologies: The interview results suggested that some aspects of modern technologies might be even vague for IS analysts to understand. As a result, it is difficult for them to clarify the details of these technologies for clients.

"I think for me, as a business analyst, I would say it is [a little bit vague]. I think I have an advanced knowledge, but not enough for me to be able to translate the meaning to the business 100%" (Participant #8)

5- Lack of a deep understanding of the project

When clients and business analysts have a poor and ambiguous understanding of a project (e.g., scope and expectations in the project are not clear), they cannot determine the precise requirements. Also, when timeline, budget, and other resources are not well allocated at the beginning of the project, requirements cannot be identified accurately. This will develop further problems and change requests in the future. Lack of a clear understanding of the project can also lead to communication and transparency problems that were discussed earlier.

"The consequence of a poor planning is more change request. [...]" To understand the scope, this is a major thing. Because what IT understands from and what the business has in mind are completely two different things. [...] It's not just understanding the scope. The problem is the end-users sometimes think they did agree on the requirement, hoping this was going to deliver. But then they get something else." (Participant #8)

Impact of OTSS: From the data, we interpreted that by using OTSS, clients can more precisely understand a system and various expectations in a project. Many requirements are already included in a packaged solution, which supports the users better visualize the system from the beginning of the project.

Impact of modern technologies: From the data, we interpreted that the complexity of modern technologies adds further ambiguity to clients' understanding of the project.

6- Business Process ambiguity in organizations

When clients do not clearly understand their business processes in the organization, they cannot provide an accurate list of requirements to business analysts. The intended IS will be designed based on wrong and missed information. As a result, it will create problems for the organization in the future.

For ambiguous business processes, clients may add new pieces of information every day and make continuous changes in their requirements even before the implementation phase.

> *"I would say that's also a very common one, where they're trying to figure out this process, but [they say] we get back to you, and then you say okay. [Later] they do have it but it's completely different from what we discussed earlier. It changes everything […] Here's a little bit more information, and here's a little bit more. And timeline gets extended and solution starts changing every day, every week." (Participant #1)*

Impact of the modern world: According to the results, we face more and more global and multinational projects in our contemporary world compared with the past. Usually in these projects, a base platform exists, and each branch adds its unique processes to the base platform. The business process ambiguity complicates the projects and IS changes.

7- Conflicts between business needs and technical standards

In some domains, especially where business processes are highly integrated with digital technologies (e.g., telecommunication industry), the business requirements of users might not be thoroughly compatible with the technical standards and best practices (e.g., network standards). In these cases, IS requirements should be adjusted to become consistent with the technical standards.

"So when everyone's trying to create business requirements, then sometimes, from the end-user perspective, the requirements they are looking for could be not aligned with what the technical team suggest." (Participants #5)

Impact of OTSS: The findings demonstrated that there are structural constraints in OTSS, particularly in cloud-based solutions. When companies implement cloud-based systems, they will have limited change options in the future. Most often, these companies should follow the default configurations provided by the vendor, and they would not be able to make all their desired adjustments.

8- Changes in environmental factors

Changes in the environment impact the requirements of IS. One of the powerful change drivers is new regulations and laws (first subcategory). Companies may have to adjust their business processes according to new regulations. As a result, this will influence their IS.

"So one of the major examples of that is when for revenue recognition […], so now there was this whole new set of laws and rules and how companies need to recognize revenue when they saw something. And actually […] it was a major shift. The laws drastically changed. So because those came out, they will change request everywhere to adjust business processes and usually use a new version or a new patch of software in order to make sure that you're doing business in accordance with the law." (Participant #2)

Firms within the financial and banking eco-system may face these changes more frequently. Sometimes, they need to continue the projects based on tentative requirements until a particular law is approved.

Dynamic changes in our contemporary world is another source of requirements change (second subcategory). As a response to environmental changes, organizations may need to change their strategies (e.g., making decisions to grow and expand their market).

"The business keeps changing. It's a dynamic environment, so sometimes, at one point, maybe the business requirements make sense to all the stakeholders, but then throughout the time, because there are different events happened during the operation, then the stakeholders may come back saying well now we can't do this if

you launch the project as is. You have to make some change to accommodate our new requests. That's more like what usually happens." (Participant #5)

"The world is changing, and their business is changing. What you've documented, the requirements, and built […] the world changed them. And things are different now. […] Every time the world changes for an organization, there are implications. You go multinational, or you start selling a different line of products. You start selling different services that you have not done before. […] They were in the West Coast only, now they are in the East Coast. They only did business in America, and now they're doing business in Canada. […] They may say we only dealt with English-speaking customers before, but now we're going to France, or we're dealing with Quebec." (Participant #6)

When new technologies are available on the market (third subcategory), many companies try to take advantage of them and catch up with the competition.

"If new technology comes out that offers new affordances, then by all means everybody tries to take advantage of it. You know because all the competitors are doing it. If you want to exist in the market, then that's your chance to actually go and take advantage of it." (Participants #3)

"Clients compare our software with the competitor's. […] Even before the client asks for the change request, sometimes we try to compete with the competitors and build something same or better.[…] I'm thinking maybe this competition is more dynamic and more competitive comparing to what we had maybe 10 years ago."(Participant #10)

"So if [as clients] you don't have the latest greatest coolest features, you're going to lose the market share. So that is a big source of change requests. In fact, it's almost the continual source of change requests, as if it's a company that's trying to be innovative and they want the latest greatest stuff." (Participant #6)

Impact of the modern world: In today's world, new technology features come to the market every day. They not only influence the macro-level organizational decisions but also impact the individual's expectations and demands, from both internal employees and external customers.

9- Changes in organizational factors

When organizational changes happen within a firm, clients request further changes in their IS requirements. New requirements are requested to effectively support the new business needs of the firm. Organizational changes can be an outcome of environmental changes (e.g., impact of new regulations on business processes), or they can be drawn from internal roots (e.g., employees turnover).

When new people join an organization, they may change the way certain things happen (first subcategory). As a result, IS should be adjusted based on the changes in business processes.

> *"Because people change, processes can change. Sometimes they want to change the process, they say we've been doing it this way for so many years, now we want to change." (Participant #10)*

Business processes may also be changed due to changes in business models of the organization (e.g., transformation to a subscription-based service) (second subcategory) or changes in other technologies, which are implemented simultaneously in the organization (third subcategory).

> *"This is a major trend, it isn't just a change request. It's a major transformation of a business, and it's going to drive a lot of changes in how the orders are structured, or in how people and their business processes are. Because today they don't have a subscription step in their business processes, but tomorrow they will." (Participant #2)*

Impact of OTSS: The results suggested that it is difficult for clients to change packaged solutions, particularly the cloud-based systems. This is because of the structural constraints that exist in these systems.

Impact of modern technologies: The findings revealed that when organizations decide to implement modern technology features, they may face resistance and complaints from their employees, specifically who are not interested in modern technologies. In these cases, organizations may need to add further requirements to facilitate the transition for those individuals (e.g., by adding new interface layers that resemble the former systems) (forth subcategory).

10- Changes in the IS project

Changes in the project's elements (e.g., changes in the scope, ownership, and participants) and constraints (e.g., changes in the timeline and budget) lead to changes in the requirements. For example, when employees responsible for a project are changed, new employees may bring different ideas to the project. This can change the direction of former requirements.

> *"I have been working with a client for 2.5 years, and this year they are being acquired by another company. And the person that I was working with is going to be fired. So whatever we've been working on then will be gone basically. We were planning so many different things, and now we don't know, we have something new. Now they're going to work for the parent company and then that person is going to leave." (Participant #10)*

> *"Your subject matter expert may change. So you have a new person and the new person may not know […] all these other areas needs to get impacted." (Participant #7)*

The examples we elicited from our interviewees underline the impact of changes that the firm's representatives, subject matter experts, and end-users bring to the project. However, we inferred that participants are not limited to these individuals. IS consultants, analysts, designers, and any other member from the vendor's team can also influence the requirements decisions.

Impact of the modern world: In our contemporary world, many companies may broaden the scope of their activities into other countries. This results in forming new expectations and constraints in their IS projects.

11- Improving user experience

After experiencing IS for a while, users may find it difficult to utilize the systems. For example, sometimes, there are gaps between existing business processes and technology features that cannot be tackled before the implementation (first subcategory). Also, sometimes, technology interfaces may not be as intuitive and usable as expected to be (second subcategory). Based on users' feedback, additional functionalities and design features will be prepared to fill the gaps or to improve the usability of IS. Improving user experience is a change driver that can emerge anytime during the tests or after the implementation.

> *"Either the tester or after implementation the correct user, when they start using a system, if it's not intuitive and it's not user-friendly, then that's where we all think we'll definitely need to make this change." (Participant #7)*
>
> *"If there's something extra you want the system to do, or if you want the system to do something slightly different in a particular scenario, that's not a bug. That's a change request. So that's one way that comes up, and that falls under the category of [improving] user experience. Because they're saying as a user in this particular case, I want the system to do this other thing, or I want this additional functionality. It's a change and they're trying to improve the user experience." (Participant #2)*

Impact of OTSS: According to the results, due to the high interdependency between modules and structural constraints in packaged solutions, many of change requests are not approved in these systems.

12- Changes in user expectations

Many people, by nature, would like to experience new and attractive tools frequently. Organizational employees are not exempt from this desire. They are exposed to various fast-changing technology features everywhere. Clients may not experience a specific problem with the designed system. However, when they compare the fascinating technology capabilities in their daily life (e.g., in various social network systems and mobile applications) with their current workplace systems, they would like to add those capabilities to their requirements.

[They] compare it to what they already know, and it's a human nature. Well I go into my Gmail and everything is fast and easy to handle. Then why am I in this system? […] Clients putting new requests and saying it's not how I really envisioned it. I expected this feature to be like Microsoft. I say well, that's not really Microsoft. We're not the product that works with Microsoft. They just do a different thing." (Participant #1)

"I asked him: How do you feel about the system? Why don't you use it? He said he hates it, and it's clunky. But, we're following the standards. We cannot make whatever changes they asked. Because, otherwise it's going to change for everyone. And he was comparing our products with iPhone or other projects, and he was saying right now the technology could reach so far. How come you cannot make whatever I am asking for?" (Participant #10)

Impact of the modern world: Our interview results showed that user expectations have significantly increased in our contemporary world. This has happened mostly due to the availability of fast-changing technologies on the market and users' exposure to these technologies, both in users' working life and personal life.

"It's all about pleasing each end-user [and their] expectations are so getting higher and higher." (Participant #1)

"I think [the expectation] is higher, because the difference now is we're all on our mobile device, iPads, tablets, and everything 24 hours a day. And you expect when you go to your office environment, your software at your office functions like [them], and that's not realistic either." (Participant #4)

13- Changes in the technical knowledge of users

Besides new technology-related information they receive from the environment every day, organizational users gain new technical knowledge continuously as they work with the intended IS (e.g., users learn more about technological affordances provided by the IS). Users may not have a clear understanding of the intended IS at the beginning of the project, but their knowledge evolves. Based on their new knowledge, users update their requirements. For example, some users agree to have a multimedia tool in their system. However, they may discover the actual power of a multimedia feature after they experience it. Then they will request additional related features on top of the basic one.

> *"This is the thing that they will never be able to know 100% what they want. So we should be ready to any change request." (Participant #8)*

Impact of OTSS: Because of the structural constraints and reconfiguration limitations that exist in OTSS, particularly in cloud-based systems, not all the change requests of users can be accepted.

14- Changes in interconnected systems

The intended IS may be dependent on other systems. If the dependee systems change over time or if they are not compatible with the intended IS, some IS requirements should be changed (first subcategory).

> *"Initially we had one system that our reporting could pull the data from. And then in the middle of our project, this system itself got upgraded or got consolidated with another system. So the new environment was completely different from the old one. And I had to submit a change request because now the data source is completely different" (Participant #5)*

Sometimes, during the process of IS development, a project may depend on a supplier (e.g., to provide a device, API, or any other tool to the project). However, the supplier firm may not be able to accomplish the job at the expected time, or it may prepare a different tool. As a result, the intended IS will go through further requirements changes according to the new situation (second subcategory).

> *"The changes were coming because you and your client were also dependent on another party. They give you some input to integrate with your system, and your expectations with what they actually have for you [are] kind of different. [...] So we had to build another layer on top of that project. This was something that had not been anticipated." (Participant #3)*

Impact of OTSS: When clients purchase a packaged solution, especially a cloud-based system, it is difficult and sometimes even impossible to change the system. Therefore, changes in interconnected systems can result in serious problems and inconsistencies in the project.

Discussion

In the previous section, we presented 14 categories of causes of requirements change in contemporary IS development projects. Except the last category — changes in interconnected systems — the others are very similar to the change drivers that were previously identified from the literature (Table 2.2). Similar to the first 13 categories, we believe changes in interconnected systems is not limited to contemporary technologies or modern IS development practices. It can emerge in other IS projects as well. Overall, we discovered that change drivers of requirements are not different in classic and contemporary IS development projects.

However, the interview results demonstrated that the modern world, OTSS, and modern technologies possess unique characteristics that moderate the processes of requirements change. We briefly explained these characteristics at the end of each change driver category. Here, we elaborate them in more detail. They form the main differences in requirements change between classic and contemporary IS development projects.

Characteristics of the modern world: The findings revealed that fast-changing technologies and globalization are two prominent characteristics of the modern world, which impact the change requests of managers and end-users in the organization. They seem to increase the number of change requests as well. More specifically, new technology features come to the market every day, and clients are exposed to these features everywhere, both in their personal life and working life. If clients become interested in new technology features, they include these features in their change requests.

> *"And he was comparing our products with iPhone or other projects, and he was saying right now the technology could reach so far. How come you cannot make whatever I am asking for? [...] We have to keep up with the competition. If we lack behind, we cannot survive in this business. There are a lot of competitors that have good products, you know, so we got to make sure we stay on top of everything not just*

for current functionalities. We need to keep adding more and more." (Participant #10)

Also, many companies have expanded internationally and opened branches in several locations. Each branch has its own culture, rules, language, and business processes, so it requires unique additional requirements.

"When you are a global company and you have multiple branches, departments [...] you know every office, every branch, every subsidiary has its own process. And the problem with that or the challenge is when you're trying to bring that all together into one platform. Yeah, there's so many, how do you make everyone's requirements at the [same time]? Sometime it's taking too many things and trying to fit the square into the circle type of thing." (Participant #1)

In sum, the above mentioned two characteristics of modern world moderate some of the classic change processes in requirements in that they can increase the impact of some change drivers (e.g., "changes in environmental factors", "changes in user expectations", and "business process ambiguity") on requirements change.

Characteristics of OTSS: Modern systems development practices, especially cloud-based solutions, significantly impact the change requests of clients. From one side, OTSS include prepared functionalities and already built requirements. Also, they have user-friendly and intuitive interfaces. These characteristics help users easily understand the product and get involved in the requirements gathering process (or customization process). In other words, using packaged solutions mitigates the impact of some change drivers (e.g., "lack of involvement of correct clients" and "lack of a deep understanding of the project"). It should be noted that vendors update and improve their packaged solutions continuously.

"For us [working with off-the-shelf solutions], the implementation time is shorter, because we have 80% of the solution and we customize 20%. I'm just generalizing. Sometimes it is 70% and 30%. It depends, or 50% and 50%. But the other company [building everything from scratch], they customize everything. Sometimes it can take 2-3 years to design and build. Wow! But for us maximum 3 months or on average 1-2 months." (Participant #10)

"Because the reality is the modern solutions have very simple interfaces and icons. You don't have to scroll through menus. There's a little icon that is for your human resource, for your vacation, and for your time off, for your pay cheque. And all those things are made through icons" (Participant #4)

From the other side, however, OTSS bring their complexities and structural constraints to requirements change processes. Particularly, interview results underlined the difficulty of making changes in cloud-based solutions. There are many interdependencies between various modules and features in OTSS, which limit the future changes in these systems. Our interviewees strongly recommended that clients keep original features made by vendors and only consider minor reconfigurations. Otherwise, systems are not compatible with future upgrades provided by the vendors. In addition, in cloud-based systems, any change in one client's system immediately becomes effective in the systems of other clients. This is why IS analysts only approve the change requests that benefit all of their clients.

"Most clients take the majority of out of box functionalities, take it exactly as the software company attended to and only make very minor configuration decisions, cause otherwise it will be challenging to continue to accept their upgrades" (Participant #4)

In sum, the above mentioned three characteristics of modern systems development practices moderate some of the classic change processes in requirements in that they can increase the impact of some change drivers (e.g., "changes in organizational factors" and "changes in user expectations") and decrease the impact of some others (e.g., "lack of involvement of correct clients" and "oversight of details and recalling issue") on requirements change.

Characteristics of modern technologies: Using modern technologies also influences the changes in IS requirements. Some modern technologies, such as data analytics and machine learning, are complicated to understand for many users and even IS analysts. This increases the ambiguity in communications and planning. In other words, using modern technologies can increase the likelihood of some change drivers (e.g., "inability to articulate the requirements” and “lack of a deep understanding of the project”). As clients gain a deeper understanding of modern technologies, they may change their requirements gradually.

> *“So for them is a little bit vague, and I think for me as a business analyst, I would say it is. I think I have an advanced knowledge, but not enough for me to be able to translate the meaning to the business 100%. Even for IT yet. So as you said it's a new technology, I would say it's not 100% new, but now this is the competitive advantage if someone is good in these technologies.” (Participant #8)*

In addition, organizational users sometimes resist accepting modern technologies (e.g., some senior users may be more resistant compared with young users). In these cases, project managers need to consider adding further requirements to facilitate the transition process for users.

> *“[In modern technologies] I think there are definitely new changes that may come in with the use of these new technologies. Users may not be familiar with the new technology, right? Or users who are used to do things in a certain way, they're resistant to change. Therefore you might need to sometimes introduce some sort of a different requirement to cater to them”(Participant #7)*

In sum, the above mentioned two characteristics of modern technologies moderate some of the classic change processes in requirements in that they can increase and complicate the impact of some change drivers (e.g., “inability to articulate the requirements” and “changes in organizational factors”) on requirements change.

Table 3.3 summarizes the mentioned characteristics of contemporary IS development projects. We also identified which change drivers are more likely to be impacted by each characteristic.

Further data would be required to verify the presumptive results. To ensure the rigor and increase the validity of our findings, another source of data would serve to triangulate with the interview data. We conducted a set of qualitative surveys, essentially to examine the impact of OTSS and modern technologies. Using multiple methods avoids the inherent limitations and potential biases of the interviewing. Details of the surveys are explained in the following section.

Table 3.3 Characteristics of Contemporary IS Development Projects

Characteristic Scope	Characteristic Name and Description	Informants	Applicable to these Change Drivers
Modern world	***Fast-changing technologies***: In our modern world, new technology products and features come to market every day, and clients are exposed to these technologies 24/7, both in their personal life and working life.	Participant #1, Participant #2, Participant #4, Participant #6, Participant #8, Participant #10	Change driver #8, Change driver #11, Change driver #12
	Globalization and multinational companies: In our modern world, many companies have various branches in different countries. Each office may have its own culture, language, rules, processes, and so on.	Participant #1, Participant #6	Change driver #2, Change driver #5, Change driver #6, Change driver #8, Change driver #10
Modern systems development practices (or OTSS)	***Predefined requirements***: OTSS already include a wide spectrum of features and capabilities. According to our interview results, OTSS may have between 50% and 90% of clients' requirements available there.	Participant #4, Participant #10	Change driver #1, Change driver #2, Change driver #3, Change driver #5
	Visualization power: Since the OTSS are tangible for users from the beginning of a project, even non-technical users can easily understand the products and get involved.	Participant #4	Change driver #1, Change driver #3, Change driver #5
	Complexity and structural constraints: In OTSS, there are many interdependencies between various modules and features, which limit the changes in these systems. It is difficult to make changes in OTSS. Only minor reconfigurations are suggested. Particularly, this characteristic is very important in cloud-based solutions.	Participant #4, Participant #7, Participant #10	Change driver #7, Change driver #9, Change driver #11, Change driver #12, Change driver #13, Change driver #14
Modern technology features	***Complexity to understand***: Modern technologies, such as big data, machine learning, and algorithm-based tools, can be difficult for users and even IS analysts to understand.	Participant #8	Change driver #3, Change driver #4, Change driver #5, Change driver #9, Change driver #11

Characteristic Scope	Characteristic Name and Description	Informants	Applicable to these Change Drivers
	Culture and resistance: Different users have different willingness and receptivity to modern technology features. For example, young users may be more interested in experiencing modern technologies compared with senior users.	Participant #7, Participant #8	Change driver #9, Change driver #11

Surveys

Data Collection

We collected data from nine IS consultants, including business analysts, systems analysts, performance analysts, and one SAP[7] support analyst, from four different firms. They had sufficient experience in the area of requirements change in contemporary IS development projects, particularly in manufacturing, distribution, mining, telecommunication, media services, and health care domains.

Except Participant #1, who is also an IS scholar, we selected our other participants by the following strategy. We contacted the chief information officers and information technology executives in three organizations (from the telecommunication, manufacturing, and government sectors) and requested that they share our invitation letter with their IS consultants (see Appendix D). From these firms, a total of eight IS consultants, whose professional background matched with the criteria in the invitation letter, volunteered to participate in our study. Table 3.4 demonstrates the background information of our participants.

Using the *Qualtrics* platform, we designed an online questionnaire, including 14 sets of questions. Each question set focuses on one of the change drivers identified from the interviews and investigates the impact of OTSS and contemporary technologies on each

[7] SAP is an acronym for Systems, Applications and Products. SAP is one of the leading software companies specialized in enterprise application software.

change driver. Each question set includes multiple-choice and open-ended questions (see Appendix E).

Table 3.4 Background Information of Survey Participants

Participant ID	Work Experience	Company/ Project Type	Answering Choice
Participant #1	40 years, Consultant, IS scholar	Consulting firm	All questions
Participant #2	20 years, Information technology business systems analyst	In-house consulting team	First group of questions
Participant #3	30 years, Business analyst, Systems analyst, and Business systems director	In-house consulting team	First group of questions
Participant #4	5 years, Senior performance analyst	In-house consulting team	First group of questions
Participant #5	15 years, Senior business systems analyst	In-house consulting team	Second group of questions
Participant #6	5 years, Business systems analyst	In-house consulting team	Second group of questions
Participant #7	3 years, Information technology SAP support analyst	In-house consulting team	Second group of questions
Participant #8	15 years, Senior performance analyst	In-house consulting team	Second group of questions
Participant #9	18 years, Business consultant	In-house consulting team	Second group of questions

Based on the feedback we received from Participant #1, we made the following two adjustments in the questionnaire. First, 14 questions seemed to be very long for our participants. So, we divided the questions into two smaller questionnaires. The first questionnaire covers the change drivers from #1 to #7. These change drivers mainly explain the requirements changes that emerge before implementation. The second questionnaire covers the change drivers from #8 to #14. These change drivers mostly explain the requirements changes that can happen at any point in a project. Each questionnaire is expected to take about 40 minutes to complete. Before sending the survey link to our participants, we asked their preference between the two groups.

Second, we realized that the term "modern technology" is not a proper label for all the mentioned technologies. Particularly, e-commerce and mobile systems have been around for a much longer time compared with machine learning, data analytics, and blockchain. Therefore, we divided our contemporary technologies into two groups. The first group includes e-commerce and mobile systems, and the second group includes the most recent and modern technologies, such as machine learning, big data, and blockchain. We then asked our participants to choose a group that they had more experience in. However, they were welcome to choose both groups if they preferred.

Findings

Survey results did not reveal any new change driver category. Only Participant #1 emphasized on the role of organizational power in changing IS requirements. Organizational power does not seem to be a new change driver, but it can be a new subcategory for "changes in organizational factors" and "changes in the IS project".

Regarding the impact of OTSS and contemporary technologies on the 14 change drivers, data revealed the following results.

1- Lack of involvement of correct clients

We inferred that clients' involvement is essential in OTSS. This is because if clients do not properly participate in the process of requirements gathering, they cannot specify all their business needs at the beginning of a project. Consequently, they will need further customization and change in the future. Customization in packaged systems is too difficult and expensive.

> *"[Off-the-shelf packages] require greater customer involvement to integrate into existing business processes, and tend to lead to higher change management. Off-the-shelf packages can be limiting in their off-the-shelf configurations, therefore*

requiring custom code which is expensive. And if lack of customer involvement, then can lead to even higher costs." (Participant #2)

For both groups of contemporary technologies, results showed that end-users' involvement is critical. Particularly, we discovered interesting findings about mobile and e-commerce systems. We discovered that end-users of mobile and e-commerce systems are not limited to internal employees of organizations, but they also include external customers of organizations. These customers come from diverse technology backgrounds. Some of them may not have sufficient technical skills, so they cannot successfully work with mobile applications and e-commerce systems. Therefore, it is important to consider the expectations and requirements of external customers at the beginning of a project as well. Otherwise, missing their requirements will result in future problems and requirements changes in the project.

"Both mobile and e-commerce solutions are targeted for a wider audience, sometimes not as technically advanced as general office/computer workers." (Participant #3)

2- Oversight of details and recalling issue

According to the findings, in OTSS, it is critical to collect all the requirements at the beginning of a project. Otherwise, clients will need further changes in the future. As mentioned earlier, change in OTSS has significant operational costs. When some requirements are not specified at the beginning, organizations usually have two options to address the missed requirements. First, they should go back to the same vendor to ask for additional services. Second, they should think about new systems (and new vendors) to address the missed requirements.

"In the future, once they need that requirement because it is an operational cost in time or money spent, then the business will need to spend more time and money either 1) going back to the vendor to see if they have an add-on service, way to customize it from their end, 2) reviewing other systems to purchase and then spend time migrating processes, data, etc." (Participant #4)

Regarding contemporary technologies, the results demonstrated that the second group of contemporary technologies evolves quickly (e.g., Participant #4 explicitly referred to artificial intelligence). This results in further complications for IS consultants and project managers to consider all the details of requirements when dealing with modern technologies. Also, we discovered that in IS projects that utilize modern technologies, requirements change at a faster pace compared with other IS projects.

> *"Because these tools evolve and update much quick […], trying to consider all aspects is almost impossible so the change requests will continue to come in and likely at a fast pace." (Participant #4)*

3- Inability to articulate the requirements

We inferred that communication between users and IS analysts can be more challenging and difficult in customizing/implementing OTSS compared with designing systems in-house. When users communicate with IS analysts from a third-party vendor (in OTSS), they may experience more difficulties to articulate their requirements as each group uses different terms and words. However, users can better articulate their requirements when they talk with internal analysts from their own organization.

> *"When you have the ability to custom a system more internally vs. an off-the-shelf system, the terminology that would typically be considered standard can be different. This can cause a disconnect when a user is trying to explain to a project lead/developer what they want. For instance, an end-user saying "field" could mean something different to the developer and they would end up modifying the wrong thing even though technically they completed the job as asked and as designed." (Participant #4)*

For the first group of contemporary technologies (e.g., mobile and e-commerce systems), the results revealed that communication between users and analysts is challenging. As mentioned earlier, external customers of organizations shape an important group of end-users in mobile and e-commerce systems. Communication between external customers and analysts can be difficult, as many external customers may not be familiar with the technical terminology.

"The communication has to be catered to a wider and sometimes less technology advanced audience." (Participant #3)

Also, the second group of contemporary technologies seems to bring additional challenges to the communication process. Since modern technologies evolve rapidly, new terms and words emerge every day. Sometimes even business analysts do not know all the terminology.

"There is so much new terminology being injected into this world that a lot of individuals are unaware of including myself. In addition, a lot of words that may have been used in the past have been rebrand, redefined making this even more confusing for new technologies that continue to evolve faster than ever." (Participant #4)

4- Lack of transparency from business analysts

Among the four participants who answered the first questionnaire, half of them did not agree with this change driver. It was probably because they had not experienced this change driver in their own projects. Also, those who agreed with this change driver did not provide us with a further explanation about the impact of OTSS or contemporary technologies on this change driver.

5- Lack of a deep understanding of the project

For OTSS, the results revealed that it is very important to define the scope and expectations at the beginning of projects.

"Expectations and deliverables/scope must be set up front and agreed to." (Participant #3)

Regarding the second group of contemporary technologies (e.g., data analytics and artificial intelligence), we discovered that planning the projects that utilize modern technologies is more difficult than planning the projects that use traditional technologies. This is because modern technologies evolve quickly and are somehow unpredictable. Lack of a proper planning at the beginning of projects will lead to further problems and changes in the future.

Particularly, it is difficult for project managers to apply waterfall IS development methodology when dealing with modern technologies. The ambiguity and uncertainty in modern technologies make the planning phase in waterfall methodology more complicated.

> *"Some of the newest technologies are so new. Many organizations still work through projects in a waterfall fashion making it extremely hard to "plan" for technologies that continue to evolve at a rapid rate. It makes it difficult to secure proper funding for the project as the costs will likely be subject to change based on how the technology evolves and assumptions we miss considering based on the little knowledge surrounding the new product. Then this becomes a cycle because without the investment, the team is less knowledgeable on the new system/product and cannot learn and make good assumptions for future projects." (Participant #4)*

6- Business Process ambiguity in organizations

The findings did not show any significant impact of using OTSS on this change driver. For the first group of contemporary technologies (e.g., mobile and e-commerce systems), the results demonstrated that external customers of organizations may not completely understand the business processes related to them. If this ambiguity cannot be tackled, it will generate additional problems in the future.

> *"In these areas, you have many more first time/new users that may not understand the system. So all potential options and possibilities must be taken into consideration when designing the system functionality." (Participant #3)*

7- Conflicts between business needs and technical standards

The survey results revealed that it is difficult for IS consultants and organizations to align business needs and technical standards in OTSS.

> *"Off-the-shelf systems are limited because we don't know what the vendor can build for us until we ask them. If the system is customizable internally, it is easier to engage these members at the initial discussion. In addition, off-the-shelf systems in general are difficult to customize so you tend to say no to a lot of change requests. However, the requests don't stop coming in and eventually you regularly make the decision to decide to find other ways to build it." (Participant #4)*

"During implementation, it is hard for business resources to fully understand technical standards required by the new software. This is when software owners need to make sure that the knowledge is passsed to the business analyst of the project"(Participant #3)

According to our findings, aligning business needs and technical standards is not only difficult in implementing OTSS, but it also is complicated when organizations design their systems in-house (e.g., when internal teams do not have the proper capability to deal with technical standards).

"Home-baked software may not pay as much attention to those topics." (Participant #1)

8- Change in environmental factors

The findings demonstrated that, most often, OTSS outperform custom-built systems when dealing with changes in the environment. This is because usually environmental factors influence a large group of organizations not only one organization. When environmental change drivers emerge (e.g., hacking attacks and security concerns), OTSS providers rapidly improve their systems based on new demands in the environment. However, when organizations decide to design new functionalities in-house, probably the results will not be as successful as the results of packaged solutions. This can lead to further problems and change requests in organizations later. For organizations that live in a dynamic environment, it can be a reasonable option to consider OTSS rather than custom-built systems.

"I am thinking mostly about security issues as an environmental factor here, and changes in this environment (hacking attempts, etc.) have driven some change in our work. [OTSS] are more vulnerable due to their visibility but on the other hand generally pay more attention to security than in- house systems. The work to secure your own products and systems is significant." (Participant #6)

The data did not reveal any significant impact of contemporary technologies on this change driver.

9- Changes in organizational factors

Although all the six participants agreed with this change driver, we could not acquire further information on the impact of OTSS and contemporary technologies on this change driver.

10- Changes in the IS project

The results demonstrated that, if clients decide to change some of their requirements due to changes in a project, many of their change requests cannot be approved. This is because of structural constraints and limitations that exist in OTSS.

> *"Any system or project changes that are requested must deal with the capabilities and limitations of [OTSS], and this is something clients can reasonably accept. The argument is much more difficult with in-house systems which they might expect you to modify to suit them."(Participant #6)*

We also discovered that, in OTSS, change in a project's constraints (e.g. budget constraints) is complicated.

> *"With vendors we are contractually obligated. If it comes to budget cuts, there is nothing that can really be adjusted with vendor costs. You may cut the scope, but you may not get back that money. This will differ from contract to contract."(Participant #5)*

Besides, we inferred that there is an interaction between this change driver and characteristics of mobile systems. Specifically, the findings showed that since mobile systems are widely used by many people on different devices, they should be compatible with various operating systems, Internet browsers, and other technology infrastructures. In projects that utilize mobile systems, when expectations and constraints of projects change, IS analysts may need to consider further requirements changes to address the compatibility concerns.

> *"The variety of OSs, browsers and versions that must be supported for mobile can be challenging." (Participant #6)*

11- Improving in user experience

The results suggested that it is very difficult to change OTSS when this change driver emerges. It is because of the reasons mentioned in the previous change drivers.

"In my experience, [when] working with vendor solutions, it has not been easy to get any changes to their out of box products. It is not a good business model for the vendor to have many variations of the product to support and maintain, so they generally refuse customer requests for changes if it is only requested by one customer." (Participant #5)

"System limitations for out of the box capabilities may not be able to support a given business requirement, unless custom code is needed. A different approach to meeting the requirement may need to be accepted." (Participant #9)

The findings showed that this change driver is popular in projects that utilize the first group of contemporary technologies (e.g., mobile and e-commerce systems). Many of organizational users are exposed to these technologies outside the workplace. This helps users quickly learn these technologies and continuously come up with new ideas and change requests.

"Online shopping is evolving, and our business partners are learning quickly as they are becoming more adept at online shopping in their daily lives. They have more opinions and ideas when it comes to commerce." (Participant #5)

Also, our findings demonstrated an interaction between this change driver and characteristics of mobile systems. Similar to what discussed in the previous change driver, when IS requirements are changed to improve user experience, IS analysts may need to consider further changes to address the compatibility concerns in mobile systems.

12- Changes in user expectations

The findings revealed that it is very difficult to change OTSS when this change driver happens. It is duo to reasons mentioned in the previous change drivers.

"Usually costs much more and takes more time to get a change done by a vendor." (Participant #5)

We could not discover any significant impact of contemporary technologies on this change driver.

13- Changes in the technical knowledge of users

The results revealed that, if clients decide to change some of their requirements because of changes in the technical knowledge of users, it is very difficult to accept requirements change in OTSS. This is due to structural constraints and limitations of OTSS.

> *"Off-the-shelf systems are not easily customized later […]. Generally, there are long-term costs associated with vendor solutions aside from maintenance. Licensing, subscription fees, etc. Vendors typically take much longer to turn around business requests as there is much more process involved in delivery from their side." (Participant #5)*

> *"Using [OTSS] products increases the likelihood that the client will appreciate these challenges through their own experience or willingness to accept the capabilities of the systems involved." (Participant #6)*

The data demonstrated that change in the technical knowledge of users happens frequently for the first group of contemporary technologies (e.g., mobile and e-commerce systems). This is because users also have the opportunity to work with similar functionalities in other systems outside their workplace. As a result, they more quickly improve their technical knowledge and come up with change requests.

> *"Business users start to understand new functionality available on the market as they use other shopping sites online. These learnings generally translate into new requirements on our commerce applications so that it stays current in the market place." (Participant #5)*

14- Changes in interconnected systems

Regarding OTSS, the results demonstrated that, most likely, changes in connected systems are not compatible with vendors' solutions. Because of structural constraints of OTSS, changes in connected systems can result in further risks for projects.

"We don't know if the change in the connected system will be compatible with the vendor solution... major risk" (Participant #5)

From the findings, we inferred that e-commerce systems are dependent on many back-end services. If any change in those back-end services happens, the main e-commerce system should be adjusted.

"The commerce system relies on many backend services and applications to run properly. If any of those change, we need to adjust the system to enable the communication between components of the system."(Participant #5)

Also, findings showed an interaction between this change driver and characteristics of mobile systems. After IS requirements are changed due to changes in interconnected systems, IS analysts may need to consider further changes to address the compatibility concerns in mobile systems.

Discussion

Survey results confirmed the characteristics of OTSS that were previously identified by interviews. For example, survey findings showed that OTSS include already built requirements, which can create a very good starting point for clients to articulate their additional requirements. Also, findings emphasized structural constraints and complexity of OTSS, which significantly limit requirements change after implementation. Particularly, this characteristic is critical for cloud-based systems. Implementing changes in OTSS is very expensive and time-consuming. This can make organizations consider other alternatives (e.g., purchasing or designing new systems) to address their remaining business needs.

In addition to confirming interview results, survey findings explored new insight into the impact of OTSS on requirements change. We discovered that while using OTSS can support users better articulate their requirements, it can generate problems for communication between users and analysts. Specifically, we inferred that when users communicate with a third-party vendor, they may not be able to successfully articulate their needs due to different terminology and language used by different organizations. It should be noted that this challenge does not seem to be exclusive to OTSS but relevant to any systems designed by external IS analysts and consultants.

Regarding the first group of contemporary technologies (e.g., mobile and e-commerce systems), we discovered new insights into the impact of these technologies on requirements change. First, the findings showed that end-users of mobile systems and e-commerce systems are not limited to internal employees of organizations, but they also include external customers of organizations. It is very important to consider the later group of end-users and their requirements as well. Particularly, it is essential to remember that some customers may not be able to work with mobile and e-commerce technologies. This group of end-users can increase the likelihood of emergence of some change drivers, such as “lack of involvement of correct clients”, “inability to articulate the requirements”, and “business process ambiguity”. If this group of end-users is overlooked in the process of requirements gathering, further problems and change requests will show up in the future.

Second, the findings demonstrated that various mobile systems and e-commerce systems are widely used by users outside the workplace. Therefore, users can quickly learn these technologies and develop new expectations. As a result, they may come up with change requests frequently.

Finally, the findings revealed that users of mobile systems use different operating systems, Internet browsers, and technology infrastructures on their mobile devices. IS utilizing mobile technologies should be compatible with various operating systems, Internet browsers, and technology infrastructures. Anytime a business requirement is changed/added, IS analysts should consider potential further requirements changes at a technical level to address the compatibility concern.

Regarding the second group of contemporary technologies (e.g., data analytics, machine learning, artificial intelligence, and blockchain), only three of the participants answered questions related to this group. Others chose to answer questions related to the first group of contemporary technologies. Our findings confirmed the first characteristic of modern technologies that was previously identified by interviews — “complexity of modern technologies”. Specifically, the findings demonstrated that modern technologies are still new and somehow ambiguous for many organizations and even IS consultants. Also, these technologies change so quickly. As a result, it is difficult for clients and IS consultants to elaborate on all the details of a project and requirements at the beginning of the project.

Conclusion

Research Summary

In this study, we investigated the root causes of requirements change in contemporary IS development projects. Specifically, we set our lens on OTSS (e.g., cloud-based solutions) and IS projects that utilize contemporary technologies (e.g., mobile systems, e-commerce systems, data analytics tools, and machine learning tools). We approached this research from a qualitative research methodology. We conducted interviews and surveys to collect data from IS consultants and project managers working in various industries and domains.

The findings revealed 14 categories of change drivers. We discovered that these change drivers are similar to change drivers that exist in classic IS development projects. However, both OTSS and contemporary technologies include specific characteristics that moderate the impact of these change drivers on requirements change. We discussed unique characteristics of the modern world, OTSS, and contemporary technologies and elaborated on their impact on each of the change drivers.

Contributions

This study makes contributions to both research and practice. In the last few years, a large number of firms across various sectors have increasingly adapted to using OTSS and modern technologies. OTSS and modern technologies evolve very fast, and every day we observe new products and technology features come to the market. Therefore, it is very important for IS requirements literature to keep an eye on these fast evolutions and examine their impact on IS requirements (Jarke et al. 2011). Among the existing requirements change studies, we found only a few pieces discuss requirements change in the modern IS context. The results of our study contribute to the existing literature on requirements change. Specifically, we identify the change drivers in IS development projects that utilize OTSS (e.g., cloud-based and on-premise systems) and modern technologies (e.g., mobile, e-commerce, artificial intelligence, data analytics, and machine learning systems). Also, we elaborate on the unique characteristics of OTSS and contemporary technologies. These characteristics moderate the impact of each change driver on requirements change. For example, our findings demonstrate that the complexity of modern technologies leads to additional complications for project managers when they plan IS projects. As a result, this will create further problems and change requests in the future. The findings of our study can be used as a useful starting point for future research in requirements change in the contemporary context.

The results of this study also have practical implications for IS consultants and project managers. IS consultants and project managers should consider various parameters and constraints in their planning, analysis, and recommendation (e.g., the scope of a project, technology cost, timeline of a project, and future training). The findings of our study provide them with useful insights into the changing behavior of requirements in OTSS and contemporary technologies. Along with other parameters, this information could be used by IS consultants and project managers to better assess various alternatives and find solutions.

Limitations and Future Research

This study has a few limitations that provide opportunities for future research. First, our study investigates the change processes and change drivers in the contemporary IS context merely from the lens of IS consultants. To gain a deeper understanding of requirements change in contemporary IS development projects, we suggest that future studies explore opinions and experience of clients as well. For example, it would be important and interesting to investigate how business owners and end-users observe the impact of employing packaged solutions and modern technologies on their requirements change. This research direction can capture the change requests proposed by clients both during IS development process and after implementation.

Second, in this study, the majority of our survey participants preferred to answer questions about the first group of contemporary technologies, including e-commerce and mobile systems. Given the rapid growth of modern technologies (e.g., data analytics, machine learning, and blockchain), it is essential to discover how they moderate change processes in IS requirements. We recommend future research dive more deeply into the impact of most recent technologies on requirements change.

Finally, we suggest future studies shed light on the impact of agile systems development methodologies when dealing with change requests of users in contemporary IS projects. Agile methodologies, such as eXtreme Programming, Scrum, and Feature-Driven Development, have emerged to improve the response of development teams to changing requirements of users (Lee and Xia 2010; Ramesh et al. 2010; Maruping et al. 2009). It is valuable to explore how agile methodologies interact with the unique characteristics of packaged solutions and contemporary technologies. It is also important and interesting to examine the ability of agile best practices to mitigate the risks and costs associated with various change drivers in the modern context. We suspect agile methodologies would perform more successfully in dealing with communication and articulation challenges compared with changes emerging from the outside of organizations.

Chapter Four

An Integrative Examination of Change Drivers in Computer Simulation Models

Introduction

In Chapter Two, we have elaborated on various social and technical change drivers in the environment, organization, and IS projects. These change drivers are interconnected, and IS requirements changes emerge as the outcome of their interdependencies. Chapter three sheds light on requirements change drivers and processes in contemporary IS projects. However, one important question remains unanswered — how the interconnection between the change drivers in contemporary projects impacts upon the final changes in IS requirements. We believe that an understanding about the joint influence of change drivers will help project managers to foresee the outcome and create effective interventions as needed.

To examine this question, in this chapter, we focus our attention on two change drivers and one characteristic of packaged solutions in cotemporary IS projects. The two change drivers that we would like to study are “changes in environmental factors” and “changes in users’ expectations”. These change drivers have been repeatedly identified by the majority of participants in our interviews (Table 3.2). As elaborated in the previous chapter, change processes emerging from these two drivers are significantly impacted by the characteristics of contemporary IS projects. Also, these two change drivers can happen at any stage of a project, either during development or after implementation. In addition, changes in the environment and users’ expectations emerge at two different levels — macro level and individual level. Therefore, it would be interesting to study how these two change drivers interact with each other. We also would like to study the impact of “complexity and structural constraints”, as a prominent characteristic of OTSS. This characteristic has been discovered in the interviews and confirmed later by the survey results. We have explained the influence of this factor on requirements change in detail in Chapter Three.

With the aim of analyzing the joint influence of the three aforementioned factors on IS requirements changes, we answer the following research question: ***What are the impacts of the interaction among the "change in the environment", "change in users' expectations", and "complexity of OTSS" on IS requirements change?*** We tackle this question by conducting a computer simulation study.

According to Davis et al. (2007), simulation is *"a method for using computer software to model the operation of real world processes, systems, or events"* (p. 481). It has been used to "describe" the complicated systems, "explore" the emergent behaviour of systems, and "develop theories" in various science and social science fields (Davis et al. 2007). There are several simulation approaches, for instance cellular automata, agent-based modeling, genetic algorithms, and system dynamics. In this research, we implement an agent-based simulation model, because it is a proper solution to represent the emergent processes associated with an environment, agents, and interactions (Benbye et al. 2020). Agent-based simulations help us model complex systems in which heterogeneity among a group of individuals exists.[8] In this study, we model how IS users, as agents in our simulation model, interact with a dynamic environment outside the organization, and experience changes in their expectations. Next, we model how the organization selects the requirements change requests provided by its members. Finally, we explore the effects of adapting to those change requests for the organization.

To answer our research question, we define four main constructs and examine their joint influence on requirements change in our simulation model. These constructs include: **1)** ***available technologies on the market*** (used to demonstrate that technology changes happen in the environment); **2)** ***technical expectations*** (used to show that changes happen in users' beliefs and knowledge); **3)** ***degree of interdependency in OTSS*** (used to show how the complexity and structural constraints in packaged solutions influence decisions about the requirements change); and **4)** ***learning pattern of the population*** (used to explore how different learning styles of users impact changes in their expectations and overall change

[8] Compared with other economics and analytics models, agent-based simulation models more effectively support us to study complex interactions among a collection of agents. They are also powerful tools to model how system-level parameters (e.g., environmental and organizational changes) interact with individual-level factors (e.g., user changes).

requests they submit to the organization). The details of each of these constructs and their interactions are elaborated in the next section.

We then conduct two experiments to analyze the joint influence of these constructs on requirements changes. In the first experiment, we investigate the impacts of three of these constructs — degree of interdependency in OTSS, available technologies on the market, and technical expectations of users — on IS requirements change. In the second experiment, we integrate the learning pattern of the population into our simulation model and explore their joint impacts on requirements change. Finally, after performing each experiment and analyzing the results, we develop theoretical propositions based on our findings.

Model Design

Foundation

The organization we model here has an information system that is represented as a series of n requirements. The organization consists of *40* agents, who are the whole population of employees. Each agent possesses a series of m expectations. An agent can change any of its expectations as it interacts with the environment. To simplify our abstraction of change in the information system requirements, we assume that the number of agent's expectations equals the number of requirements in the information system (i.e., n=m).

The information system in our simulation model changes as the joint result of the change in the available technologies on the market and the change in the technical expectations of agents. We implement these changes in our model based on the change processes identified by our previous qualitative study. The next three subsections describe the details of our implementation. More specifically, we explain how the technology features on the market change, how agents adjust their expectations in response to these technological changes, and finally how the organization changes the information system based on new requests of agents.

It should be noted that the available technologies on the market, technical expectations of agents, and the organizational information system are respectively encoded as "information" about the available technologies on the market, technical "knowledge" of agents, and organizational "knowledge" of the information system. Since in our simulation model, we deal with changes in the knowledge of the organization and agents, we operationalize the concepts from the organizational learning literature to implement the three aforementioned change processes. More particularly, the change processes we implement here are inspired by the "Mutual Learning Model" proposed by March (1991).

Change in Available Technologies[9] on the Market

The technology features available in the environment are independent of the agent's expectations and organizational information system. In our model, technology features available on the market are specified as a series of *s* technology features. Each technology feature has a value, which defines the perceived benefit that the organization gains after implementing that technology feature. We assume that number of technology features available on the market equals the number of requirements in the organization's system ($n=s$).

To represent a dynamic environment where the technology features available on the market change continuously, we implement a technology features pool. The pool includes *q* technology features ($q=2s$). Every time we run the simulation, the environment randomly selects *s* elements from the pool and changes the technology features available on the market accordingly. The range of values includes numbers from *10* to *59*.

Changes in Users' Expectations

As explained earlier, each agent possesses a series of *m* expectations ($m=s$). Each agent also has a learning style, which is explained below. Every time agents are exposed to the technology features available on the market, they can change some of their expectations based on those technology features and their learning style.[10]

In our model, we define three styles of learning for agents — learning at a high rate, learning at a medium rate, and learning at a low rate. Every time we run the simulation, agents with the high, medium, and low learning rates can respectively change a maximum of

[9] By technologies available on the market, we refer to the "technology features" that users observe in the environment and become interested in (e.g., the features of different applications they install on their phone)

[10] In figure 2.2, we illustrate how various social and technical change drivers, from three different levels, can connect to each other. The change process mentioned here also provides a new example of how the changes in the "technology" element within the environmental context can pass through the "actors" elements in the organizational context and IS development context.

$m/2$, $m/3$, and $m/5$ of their expectations according to the technology features available on the market.

In reality, each employee may change her/his knowledge about the technology features in a different manner. For example, the interview results have revealed that young users might be more interested in experiencing and working with modern technologies compared with senior users. There are usually some users in the organization who have a higher level of resistance to experiencing new technologies compared with other users. These people prefer to continue using old technologies. In our simulation model, the three learning styles can represent the fast-learning, moderate-learning, and low-learning capabilities of IS users when they deal with modern technologies.

At the beginning of a simulation session, a learning rate is randomly assigned to each agent. It should be noted that agents have no knowledge about the value of each feature. Based on their learning rate, agents randomly choose a subset of the available technology features, but they do not make an assessment of the value of those features.

We implement this process according to the learning process of IS users in real projects. In reality, IS users have a limited attention span. As a result, they cannot accurately assess all the aspects of technology features. For example, as discovered from the interviews, some users only pay attention to the usability and performance dimensions of the technology features that they observe in the environment. However, there are so many other factors, indicating the real value of those technology features, which are missed in users' assessment. These factors include but are not limited to the total price of the technology features for the organization, training time for the employees, and level of compatibility between those features and other technology features that have already been implemented in the organization.

Changes in Information System Requirements

As mentioned earlier, the information system is defined as a series of *n* requirements. Based on the expectations of agents at each time, the organization decides to change (some of) the requirements of the information system.[11]

[11] The change process mentioned here also provides a new example of how the changes in the "actors" element within the IS development context, in figure 2.2, develop further changes in the IS requirements context.

More specifically, the organization first calculates the overall value of the *m* expectations of each agent — called the *proposed technology value* in our model. Given the range of values for each technology feature (from *10* to *59*), the proposed technology value for each agent can be a number ranging from *10s* to *59s*. Regarding each agent, this number shows the technology value that the organization will gain if it accepts the expectations — change requests — suggested by that agent.

Second, the organization compares the proposed technology value of all the agents with its current information technology value[12]. The organization chooses an agent, whose proposed technology value is greater than the organization's current information technology value. We call it the *winner agent* in our model. Third, the organization changes the current requirements of the information system to the change requests of the winner agent.

Degree of Complexity and Interdependency in OTSS

Inspired by the business researchers' adaptation of *NK Model* (Levinthal 1997), we define the complexity and internal interdependencies in packaged solutions as *K*. *This* implies that each requirement of the information systems interacts with *K* other requirements and cannot be changed independently of them. Most often, in custom-built systems, a very low degree of interdependency exists, which allows the organization to change its requirements independent of each other. However, the previous study has elaborated that in packaged solutions, especially in cloud-based systems, there is a higher level of complexity and interdependency in the systems. In these projects, the organization cannot simply decide to implement a new requirement without considering its compatibility with the whole setup supported by the vender.

In the simulation model, when *K=0*, the organization accepts all the change requests of the winner agent and adjusts the requirements of the information system based on them. However, as the value of *K* increases, the organization should first examine the K neighbors of each change request. The organization then accepts each change request only if the compatibility between neighbors is met[13].

---

[12] We define the information technology value of the organization a little later under the "outcome measures" subsection.

[13] To simplify the compatibility check inside a change requests set, we assume the following rule. As mentioned previously, each technology feature is associated with a value. Each value is a number > *0*. The

Learning Pattern of the Population

As explained earlier, there are three possible styles of learning for agents in our simulation model. One of the three learning rates is randomly assigned to each agent at the beginning of the simulation. As a result, if we apply these learning styles to a population, approximately $1/3$ of the population receives the high learning rate. Another $1/3$ of the population receives the medium learning rate, and the last $1/3$ of the population receives the low learning rate. This random distribution makes up the learning pattern of the population in the model.

In reality, however, there can be many other possible distributions of learning rates over a population of users in the organization. For example, some organizations may have a large proportion of fast learner employees, while other organizations may have a large proportion of slow learner employees. While the interview results show that some users are more interested in (or resistant to) experiencing modern technologies, the findings are not sufficient to reveal a generalizable distribution of learning styles. Therefore, rather than fixing the distribution of learning styles in our model, we decide to include the learning pattern of the population as a variable in order to explore its influence.

More specifically, in the first experiment, we use the original learning pattern elaborated in the first paragraph (i.e., assigning a learning rate to each $1/3$ of the population). In the second experiment, we compare three different learning patterns of the population to explore the influence of this construct on the requirements change.

Although we did not collect data about the impact of learning style of clients or learning pattern of client groups in our previous studies, an examination of these concepts in the current study can produce actionable implications. While IS project managers and consultants may not be able to control the emergence of the majority of change drivers, they can still design and implement training programs for IS clients to better respond to change drivers. If simulation results reveal interesting findings about the influence of learning, IS

---

organization accepts the change requests with an odd number value. For change requests with an even number value, the organization accepts them only if all the next four change requests in the set have a value with an odd number. Otherwise, the organization keeps its former information system requirement. As a result, it is very possible that the organization only accepts some of the change requests proposed by the winner agent and combines those accepted ones with some of its former requirements.

project managers and consultants can utilize these results to develop effective intervention strategies.

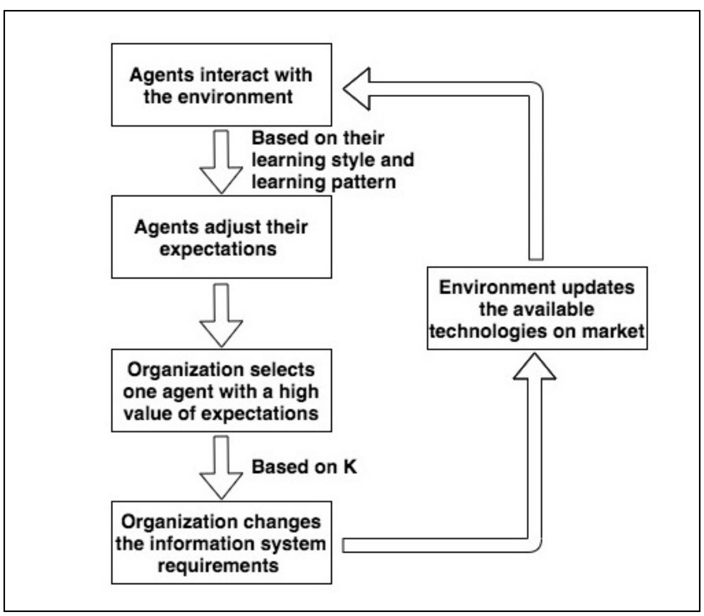
Outcome Measures

We define two outcome measures in our model. First, ***information technology value*** is the sum of the values associated with requirements of the information system at each time. It represents the overall technology benefits the organization gains after implementing the information system.

Second, ***number of requirements change*** is the number of requirements of the information system that the organization decides to change at each time. If changes are costly to implement, this measure represents the difficulty level of the implementation. However, if changes are considered as opportunities to improve the business, then this measure can also represent the level of innovation in the organization.

Figure 4.1 summarizes all the main constructs and steps mentioned in this section. These steps should be performed by the agents, organization, and environment at each iteration of the simulation. An iteration represents a point of time in reality. The model pseudo-code can be found in Appendix F. The format of this pseudo code is inspired by the pseudo-code proposed by Nan and Tanriverdi (2017). We employed *NetLogo* as the modeling environment to implement the simulation.

Figure 4.1 Summary of the Main Steps at each Iteration



Model Validation

According to Taber and Timpone (1996), three important aspects in testing the validity of a computational model are evaluating the *"correspondence between a model's output and real-world data",* evaluating the *"correspondence between a model's mechanisms and the real-world processes"*, and evaluating the *"internal logic of a model"* (Chapter 5). Except for the learning rate and learning pattern concepts, which are included in our simulation model for an exploratory purpose, all other constructs and processes in the model are drawing upon the real-world data and processes. For example, changes in available technologies on the market, changes in users' expectations, interdependencies in OTSS, and the influence of each of these factors on requirements change were all elaborated by our qualitative study in Chapter Three.

We also decomposed the model and tested each process independently. We then integrated the related processes and tested them in interaction. This helped us assure the model corresponds to real-world data and processes (Taber and Timpone 1996). For example, we tested the impact of changes in the environment on changes in expectations of agents and the impact of changes in expectations of agents on final changes in requirements separately. We then integrated these two processes. The results correspond to real-world observations, in which changes in the environment result in IS requirements change.

Experiment 1: Investigating the Influence of the Internal Interdependencies on Requirements Change

Experiment Setup

To examine the integrative impact of the two main change drivers and the complexity in OTSS, two experimental conditions are designed, one with $K=0$ and one with $K=4$[14]. In both groups, we create *40* agents having a series of *20* expectations ($m=20$). Accordingly, the environment includes *20* available technology features ($s=20$), and the information system consists of *20* requirements ($n=20$). Meanwhile, agents with a high, medium, and low learning rate could respectively adjust up to *10*, *6*, and *4* of their expectations at each

[14] We ran the simulation for $K=0$, $K=1$, $K=2$, and $K=4$. To better analyze and demonstrate the impact of a low and high internal interdependency on requirements change, we selected $K=0$ and $K=4$ as our experimental conditions.

iteration. The technology value proposed by each agent, at each iteration, could vary between *200* (=*10s*) and *1180* (=*59s*).

Each experimental condition is replicated 50 times (*sample size=50* and *total runs=100 times*), and each replication includes the execution of the model for 50 simulation iterations. Per each replication, we calculate the *average of information technology value* and *average of the number of requirements change*.

Findings

An independent sample t-test is conducted on the results of the average of information technology value and average of the number of requirements change for two experimental conditions. Figures 4.2 and 4.3 exhibit two sample results of the outcome measures for *K=0* and *K=4* groups to help the readers visually observe the general differences between two groups.

Information technology value: The group with *K=0* interdependency ($M = 1050.27$, $SD = 8.82$) is associated with a higher level of the information technology value compared with the group with *K=4* interdependency ($M = 1017.77, SD = 39.03$). The statistical analysis shows significant differences between these two groups $t(53.64) = 5.672$, $p = 0.000$.

Our findings suggest that after implementing the IS in the organization, a lower degree of interdependency allows the organization to achieve a higher information technology value. When there is no internal interdependency in the information system (*K=0*), at each iteration, the organization accepts all the change requests of the winner agent. This guarantees an increase (or no change) in the information technology value at each iteration. However, at a higher level of interdependency, the organization has less control over the final results and may even end up with a worse value. As a result, when *K=0*, the organization can gain a higher information technology value in *50* iterations. Consequently:

Proposition 1: Increased interdependency of IS requirements can heighten the challenge for organizations to gain values from the information technologies.

Number of requirements change: The group with *K=0* interdependency ($M = 2.16$, $SD = 0.90$) demonstrates lower numbers of requirement changes compared with the group with K=4 interdependency ($M = 3.97, SD = 2.37$). The statistical analysis shows significant differences between these two groups $t(62.97) = 5.006, p = 0.000$.

As expected, the results confirm that a lower level of interdependency results in higher numbers of requirements change per iteration. The findings also reveal that when there is a lower level of interdependency between requirements, the changes emerge less frequently (e.g., only through a few iterations on Figure 4.2). As the interdependency increases, the organization more frequently faces small-magnitude changes (e.g., the fluctuation pattern in Figure 4.3).

Figure 4.2 Results of One Replication (K=0)

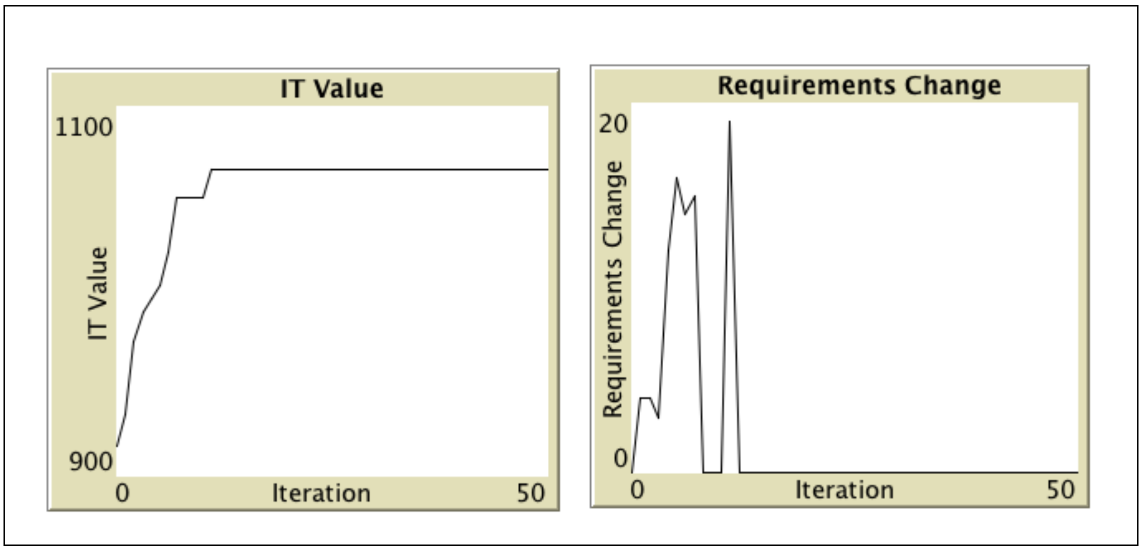
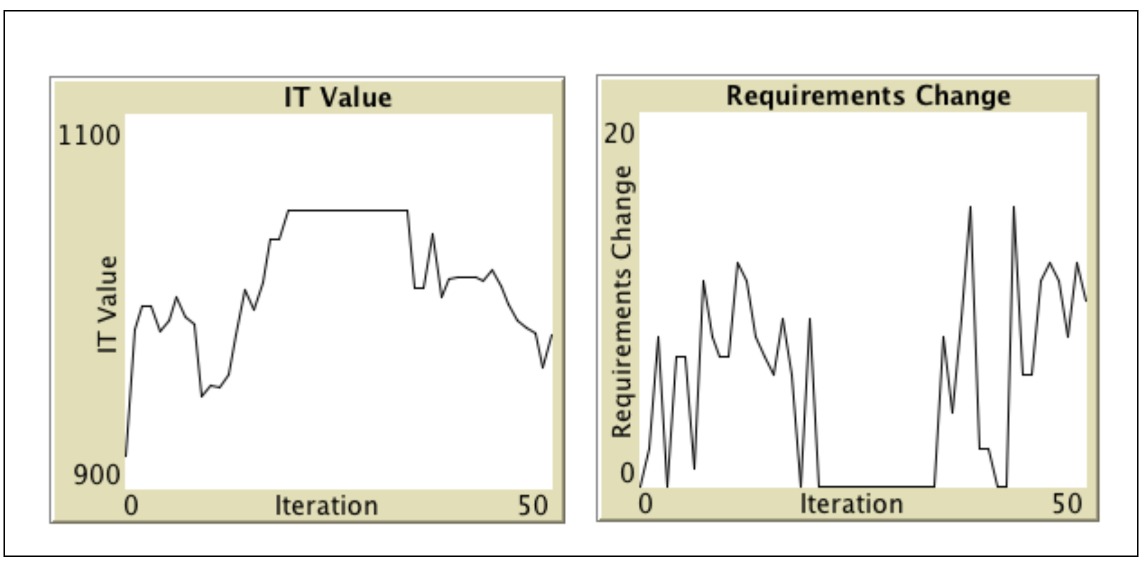


Figure 4.3 Results of One Replication (K=4)



The results suggest that when there is no interdependency in the information system, the organization implements a lower number of total changes but gains a higher total value. However, with increasing interdependency in the information system, the organization implements a higher number of total changes but experiences a lower total value.

Consequently:

Proposition 2 (a): As the level of interdependency between requirements of IS increases, organizations experience increased occurrences of small-magnitude changes in their requirements over time.

Proposition 2 (b): As the level of interdependency between requirements of IS decreases, organizations experience decreased occurrences of large-magnitude changes in their requirements over time.

Up to this point, we have examined the integrative impact of changes in the available technologies on the market, changes in the technical expectations of users, and the degree of interdependency between requirements on the requirements changes. In the next section, we include the effect of the learning patterns of the population into our model and analyze the results.

Experiment 2: Investigating the Influence of the Learning Pattern of the Population on Requirements Change

Experiment Setup

To explore the impact of the learning pattern of the population, three experimental conditions are implemented. In the first condition, we use the random distribution discussed in our base model. More specifically, $1/3$ of the population receives the high learning rate. Another $1/3$ of the population receives the medium learning rate, and the last $1/3$ of the population receives the low learning rate.

In the second condition, we implement a population with a high proportion of fast learner employees. Specifically, $2/3$ of the population receive the high learning rate. In the remainder of the population, we randomly assign either the medium or low learning rate to each agent. This group could represent the organizations that have a large group of users interested in experiencing the modern technologies.

In the third condition, we implement a population with a high proportion of slow learner employees. More particularly, 2/3 of the population are assigned the low learning rate. In the remainder of the population, we randomly assign either the medium or high learning rate to each agent. This group could represent the organizations that have a large group of users who are resistant to working with the modern technologies.

In all three conditions, we create *40* agents having a series of *20* expectations (*m=20*). Other than the learning pattern of the population, the simulation settings remain the same as before. The agents with a high, medium, and low learning rate could respectively adjust up to *10*, *6*, and *4* of their expectations at each simulation iteration. As in Experiment 1, the technology value proposed by each agent, at each iteration, could vary between *200* and *1180*.

For both *K=0* and *K=4*, each experimental condition is replicated 50 times (*sample size=50* and *total runs=300 times*), and each replication includes the execution of the model for 50 simulation iterations. Per each replication, we calculate the average of information technology value and average of the number of requirements change.

Findings

For ***K=4***, an analysis of variance (one-way ANOVA) is conducted. Table 4.1 summarizes the results. The statistical analysis does not show significant differences between these three groups, yet it reveals interesting insights into the impact of different learning patterns on the information technology value and number of requirements change.

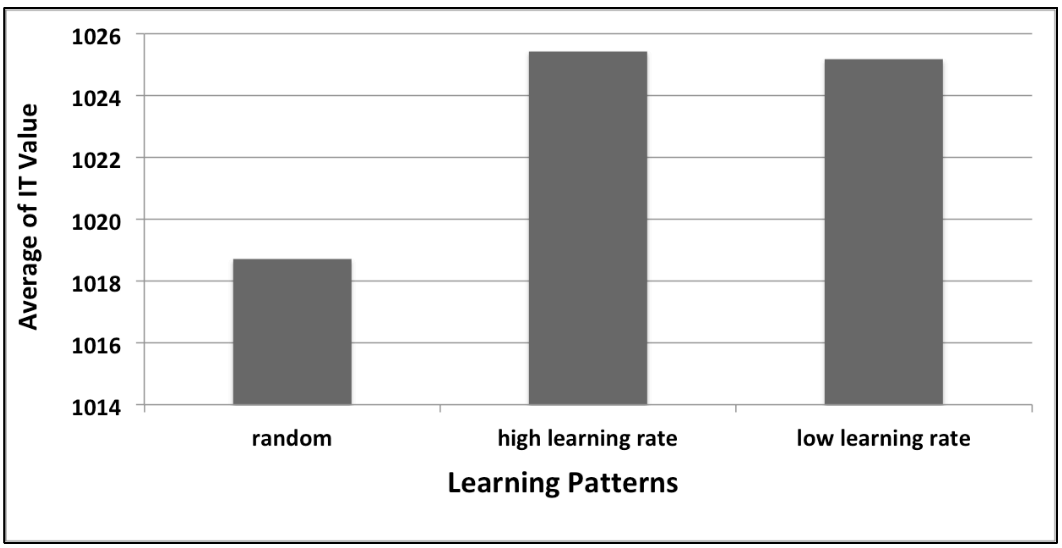
Table 4.1 Effect of Different Learning Patterns on IT Value and Number of Requirements Change (K=4)

Learning Pattern	**Average of IT Value (Mean)**	**Average of Requirements Change (Mean)**
Random assignment	1018.7108	3.9712
High learning rate	1025.4334	3.3896
Low learning rate	1025.1718	3.5960

Information technology value: As illustrated in Figure 4.4, the second group provides the organization with the highest information technology value. In this group, the majority of the agents possess the high learning rate, so overall they can experience a higher number of new technology features. Compared with the other two groups, the second group proposes the widest spectrum of change requests to the organization. As a result, the population in the second group empowers the organization to select from the change requests that propose a better value.

Before analyzing the results, we expect the third group would be associated with the lowest information technology value. However, the findings reveal that the third group provides the organization with a higher information technology value compared with the first group. This can be interpreted as the advantage of slow learning. The majority of agents in the third group cannot propose as many high value change requests as the second group. However, they can at least protect the organization from choosing the change requests with a low value.

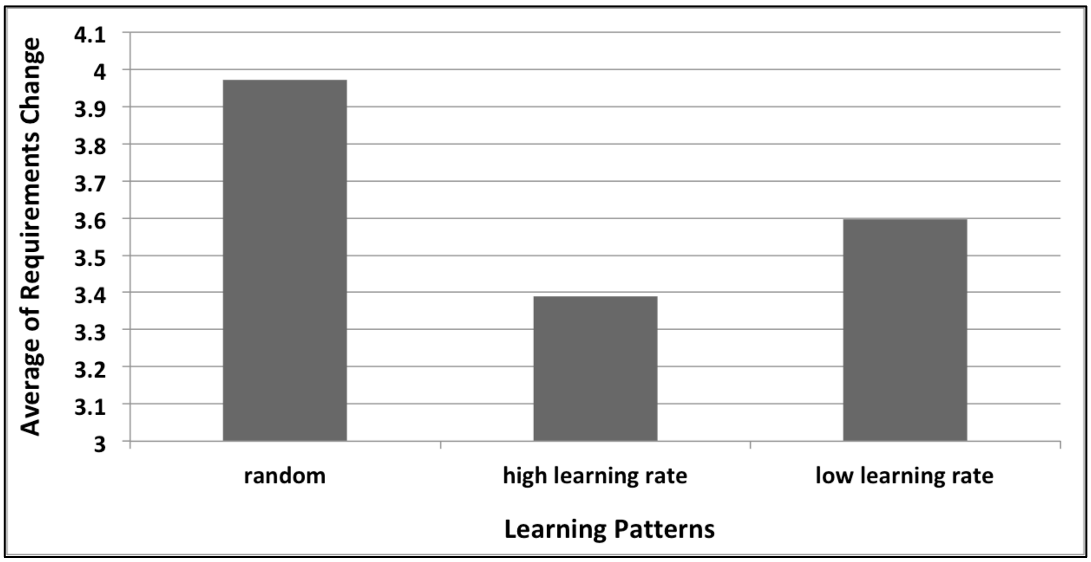
Figure 4.4 Effect of Different Learning Patterns on IT value (K=4)



Number of requirements change: The findings show that the second group provides the lowest number of requirements change to the organization (Figure 4.5). We suspect this is related to the previous analysis. In the second group, the majority of agents can experience a higher number of new technology features at each iteration. Consequently, they can provide the organization with a higher number of better change requests. Therefore, the organization can make better decisions at each iteration, and it more quickly improves its information technology values. Usually, in the states of higher information technology value, the organization receives less valuable change requests and gradually stabilizes there.

Also, the findings demonstrate that the third group provides the organization with a lower number of requirements change compared with the first group. The majority of agents in the third group are slow learner agents, who change their expectations less frequently. As a result, they provide the organization with a lower number of new change requests at each iteration. Consequently, the total number of requirements changes that the organization receives from the third groups is lower than the total number of requirements changes submitted by the first group.

Figure 4.5 Effect of Different Learning Patterns on Number of Requirements Change (K=4)



For ***K=0***, also an analysis of variance (one-way ANOVA) is conducted. Table 4.2 summarizes the results. The statistical analysis does not show significant differences between the three experimental conditions (the differences are actually smaller than the differences in *K=4*). However, again they reveal interesting and useful insights into the exploration of the impact of different learning patterns on the information technology value and number of requirements change.

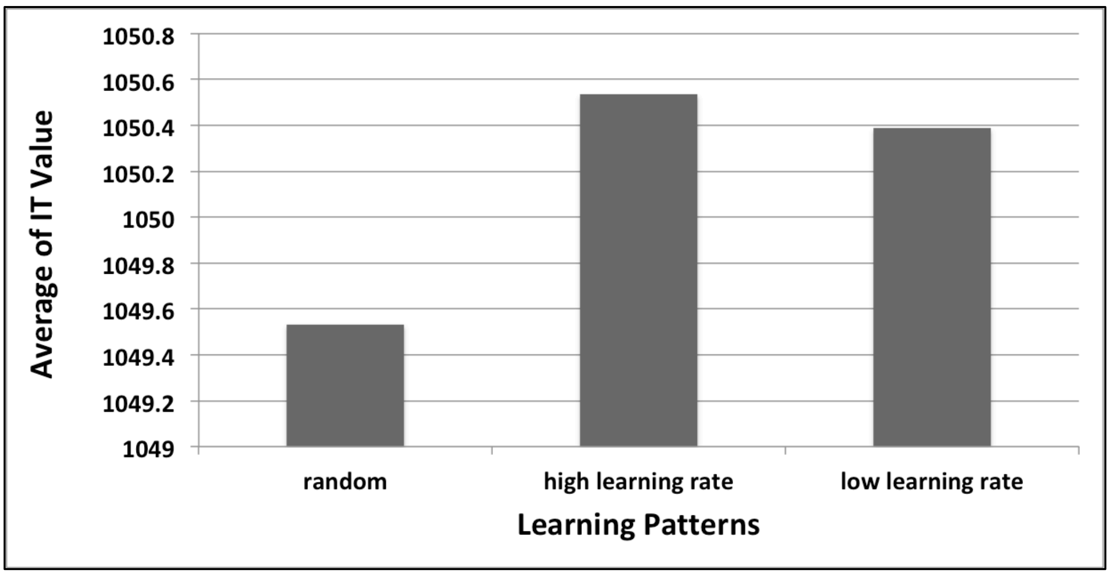
Table 4.2 Effect of Different Learning Patterns on IT Value and Number of Requirements Change (K=0)

Learning Pattern	Average of IT Value	Average of Requirements Change
Random assignment	1049.5318	2.1692
High learning rate	1050.5346	2.1832
Low learning rate	1050.3872	2.1788

Information technology value: Similar to the results of *K=4*, the second group provides the organization with the highest information technology value, and the third group is associated with a higher information technology value compared with the first group. Figure 4.6 exhibits these findings.

Moreover, a comparison between the data from *K=4* and data from *K=0* conditions confirms our previous analysis about the impact of the level of interdependency between requirements on the information technology value. Here again, the findings demonstrate that a lower degree of interdependency can lead to more opportunities of change for the organization. This in turns can support the organization to obtain a higher information technology value over time.

Figure 4.6 Effect of Different Learning Patterns on IT Value (K=0)



Consequently:

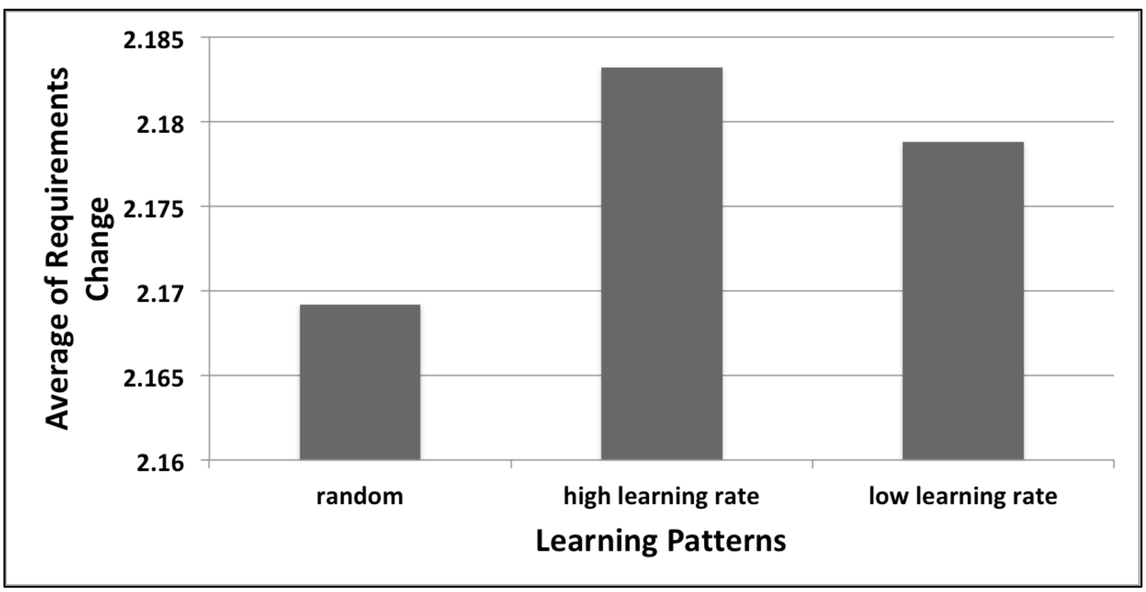
Proposition 3 (a): Organizations with a large proportion of fast learner employees gain a higher level of information technology value compared with organizations with a large proportion of slow learner employees or organizations with an equal proportion of slow and fast learner employees.

Proposition 3 (b): Organizations with a large proportion of slow learner employees gain a higher level of information technology value compared with organizations with an equal proportion of slow and fast learner employees.

Number of requirements change: We find that the second group generates the highest number of requirements changes for the organization (Figure 4.7). The difference between the three groups is very small. However, it can still reveal a very interesting impact. A comparison between the data from $K=4$ and $K=0$ groups shows an "interaction effect" between the degree of interdependency and the pattern of learning of the population, in such a way that higher interdependency reverses the influence of the pattern of learning on the number of requirements changes. More specifically, when $K=0$, the second group generates the highest number of requirements change. However, as K increases, this group generates the lowest number of requirements change.

It should be noted that the findings also confirm our previous analysis about the impact of the level of interdependency on the number of requirements change. Here again, the results show that a higher degree of interdependency between requirements results in a higher total number of changes in requirements.

Figure 4.7 Effect of Different Learning Patterns on Number of Requirements Change (K=0)



Consequently:

Proposition 4 (a): When there is no interdependency between requirements of IS, organizations with a large proportion of fast learner employees experience a higher number of requirements changes compared with organizations with a large proportion of slow learner employees or organizations with an equal proportion of slow and fast learner employees. As the level of interdependency increases, these organizations experience a lower number of requirements changes compared with other two organizations.

Proposition 4 (b): When there is no interdependency between requirements of IS, organizations with a large proportion of slow learner employees experience a higher number of requirements changes compared with organizations with an equal proportion of slow and fast learner employees. As the level of interdependency increases, these organizations experience a lower number of requirements changes.

Conclusion

Research Summary

The aim of this study is to dive more deeply into some of the findings from Chapter Two. We investigated the research question: what are the impacts of the interaction among the change in the environment, change in users' expectations, and complexity of OTSS on IS requirements change? To answer this question, we employed an agent-based simulation model to examine the integrative impact of changes in technology features available on the market, changes in users' expectations, and the level of interdependency between requirements (K) on information technology value and number of requirements changes.

The results from the first experiment revealed that a higher interdependency between requirements can increase the challenge for organizations to gain values from the information technologies over time. Also, we discovered that a higher interdependency between requirements may result in more occurrences of small-magnitude changes in requirements for organizations over time.

In addition, the findings from the second experiment provided interesting exploratory insights into the influence of the learning pattern of the population on requirements change. First, we discovered that organizations with a large proportion of fast learner employees can gain a higher level of information technology value compared with organizations with a large proportion of slow learner employees or organizations with an equal proportion of slow and fast learner employees.

Second, we explored a possible interaction effect between the degree of interdependency and the pattern of learning of the population. More specifically, we inferred that when there is no interdependency between requirements of IS, organizations with a large proportion of fast learner employees experience a higher number of requirements changes compared with organizations with a large proportion of slow learner employees or organizations with an equal proportion of slow and fast learner employees. However, as the level of interdependency increases, these organizations experience a lower number of requirements changes compared with the other two organizations. In other words, the findings showed that a higher interdependency between requirements can reverse the influence of the pattern of learning on the number of requirements changes.

Finally, drawing upon these findings, our study proposed seven theoretical propositions that could guide future research.

Contributions

This study makes several contributions to both research and practice. This chapter contributes to the existing literature on requirements change. We demonstrate that requirements changes in OTSS provide new challenges for organizations in such a way that when there is a high level of interdependency between requirements, attaining information technology value becomes more challenging. This elaboration could be very useful for assessing the risks in requirements management studies. Moreover, to the best of our knowledge, none of the existing requirements change studies have empirically investigated the impact of the structural constraints in OTSS and the learning pattern of the IS users on requirements change. The results of our empirical study can be used as a useful starting point for future research work in the requirements literature.

Our study also employs a novel methodological approach with regards to requirements change. Using an agent-based simulation method allows us to examine the integrative impact of the "changes in the environmental factors" and "changes in the user expectations", as two key drivers of requirements change, as well as the "internal interdependencies" on the information technology value and number of requirements changes. This approach complements the more commonly used qualitative or quantitative methods by providing a parsimonious model to examine qualitative insights in a quantitative way. Therefore, we could provide quantitative insights into our previous qualitative findings. Our simulation models are built on the results of our interviews and surveys. Triangulating the data from a qualitative and a simulation study provides researchers with specific advantages, which were previously elaborated by Levin and Prietula (2012): this combination enables researchers to mix the benefits of both approaches (e.g., rich data and external validity from the qualitative side and computational experiment and internal validity from the simulation side). Therefore, our study sheds light on the benefits of using triangulation by combining qualitative study and agent-based modeling to investigate the requirements change in a dynamic environment.

The results of this study also have practical implications. The findings from the first experiment suggest that it is to the benefit of managers not to purchase or sign a contract for a particular packaged solution if it is not the best fit for their business needs. This is because organizations are likely to be restricted to only minor modifications after the implementation. These minor modifications would not allow organizations to produce better results compared with the initial outcome of the packaged solutions. For instance, if an organization purchases a cloud-based system that is not a good fit for its business needs, making minor modification to the system would not yield a more desirable outcome. This also emphasizes the essential role of understanding the business requirements before selecting any OTSS.

The second experiment, which is an exploratory study, shows an interesting result. Organizations with a large proportion of fast learner employees can gain a higher value from information technologies compared with organizations with a large proportion of slow learner employees or organizations with an equal proportion of slow and fast learner employees. The finding provides useful practical implications for managers. In systems with a high internal interdependency (see Figure 4.4), managers are able to intervene to increase the technology value, despite all the restrictions of change. This intervention consists of investing in training and educating the users and providing opportunities to them for exposure to the dynamic technology market. For example, managers can send users to technology workshops and seminars where they can boost their technical knowledge. This would allow them to bring their new knowledge back into their organization and hence contribute to enhancement of the information technology value. The aforementioned suggestion is expected to be an effective solution, particularly for organizations with an equal proportion of slow and fast learner employees.

Limitations and Future Research

This study has a few limitations. One of the underlying assumptions in our proposed model is that the source of change in the information system is always within the organization. However, in reality, the periodic upgrades from vendors also change an information system over time. We also assume that change requests proposed by different users have similar weight, but in reality organizational power plays a significant role in decision-making processes. Besides, there are only two change drivers in our model. These assumptions are made to keep the model simple enough to discover the joint influence of the change drivers and the interdependency level (K). For future research, we suggest that simulation models take into account the limitations of this study and expand on our proposed model. For instance, to represent a more precise abstraction of IS development projects, it would be important and interesting to integrate the following factors into the simulation model: periodic upgrades by vendors, organizational power (e.g., change requests proposed by different users may imply different weights), and other change drivers elaborated in Chapter Three (e.g., strategy changes, human resource changes, business process changes, and changes in several IS connected to each other).

In this study, we defined three different learning styles for agents (learning at a high rate, learning at a medium rate, and learning at a low rate) and three different learning patterns for their population (population with an equal proportion of fast and slow learner agents, population with a large proportion of fast learner agents, and population with a large proportion of slow learner agents). To increase the validity of findings, we recommend that future studies expand on these categories in the simulation model and conduct further experiments by using other different numerical values for the learning style of agents and learning pattern of the population. This will help us more deeply understand the non-obvious findings of the second experiment (e.g., interesting behavior of the population with a large proportion of slow learner agents).

In sum, this study opens interesting avenues for future research. In this research, we have developed theoretical propositions to propose relationships among the constructs in our agent-based models. We suggest that future studies conduct lab or field experiments to evaluate these propositions. Particularly, it is valuable to empirically examine the relationship between the interdependency level (K) and information technology value, as well as the interaction effect between the interdependency level and the learning pattern of the population.

Chapter Five

Conclusion

Management of the *changes in requirements* is one of the most challenging aspects of IS projects. Sometimes, changes in requirements are called misfortune. Because if IS projects would employ more proper methods to communicate, determine, and document the requirements, some of these changes most likely would never happen. However, sometimes, changes in requirements are considered as great opportunities, because they can help organizations to implement their innovative business solutions. Regarding both perspectives, we believe managing the changes in requirements is often a complicated process for the majority of the stakeholders and participants in a project (e.g., IS analysts, IS project managers, business owners, external IS consultants, and third-party vendors).

In this dissertation, we shed light on the phenomenon of requirements change in IS projects. More specifically, we strived to deeply investigate the *root causes of change* and *emergent change processes* in IS requirements. This investigation provided us with interesting insights that could contribute to the extant body of knowledge in the IS requirements field. Also, our findings could have practical implications for all the IS analysts, IS project managers, business managers, and third-party vendors.

To attain our goal, in this dissertation we conducted three separate but interdependent studies:

As elaborated in Chapter Two, our *first study* elicited and synthesized various change drivers of requirements from the existing requirements literature. These change drivers were drawn upon the most recognized journals within IS and requirements engineering disciplines. They included a wide spectrum of social and technical reasons for change in requirements. For example, changes in economics, regulations and laws, organizational culture, organizational rules, and business processes are considered as some of the key social change drivers. Also, changes in the available technologies on the market, technology capacities of organizations, technology policies, and technology capabilities shape some of the important technical change drivers.

From the existing studies, however, we could not gather further information about how these change drivers are interconnected or how they jointly develop the changes in requirements. To find the answer, we employed a design science research methodology. Drawn on the studies of the socio-technical systems, we proposed the "*Socio-Technical Change Framework*". This framework integrates all the social and technical change drivers and elaborates on how they jointly shape the final changes in IS requirements. From a practical perspective, our framework enables IS analysts and project managers to attain a deep and comprehensive understanding of the logic behind the requirements changes. Particularly, this framework is applicable to all the business domains (e.g., finance, health care, higher education, retail, etc.).

As the last outcome of the first study, the *Socio-Technical Requirements Change Method* was proposed. This method demonstrates the use of the Socio-Technical Change Framework in modeling different change processes in IS requirements. Specifically, drawn upon the aforementioned framework, this method presents a new solution to validate the IS requirements and to anticipate the potential future changes in them, as much as possible. From a practical aspect, the Socio-Technical Requirements Change Method can be employed as a requirements analysis tool by IS analysts.

In sum, the Socio-Technical Change Framework and Socio-Technical Requirements Change Method are two design science artifacts. They have been founded upon previous studies. However, to increase the validity of these two artifacts, further empirical research should be performed. Specifically, we invite requirements studies to conduct field experiments or case studies about the efficiency and effectiveness of these two artifacts.

Chapter Three explained our *second study*. In this study, empirical insights into the requirements change complemented our previous theoretical findings. Specifically, we investigated the change drivers and change processes in IS requirements based on the data collected from the real world. In this research, we focused our attention on *contemporary IS projects*. By contemporary, we referred to projects that employ OTSS (e.g., cloud-based solutions and on-premise systems) or utilize recent technologies (e.g., mobile applications, e-commerce, machine learning, artificial intelligence, and data analytics). In our study, we first strived to explore the change drivers in contemporary IS projects. Next, we discovered the impact of OTSS and recent technologies on the requirements change.

We approached this research from a qualitative research methodology. We triangulated the data from two sources — interviews and surveys. The findings revealed 14 categories of change drivers in contemporary IS projects. While these 14 categories looked similar to the results of our literature review in Chapter Two, we discovered interesting impacts of the contemporary context on them. Specifically, we found that the modern world, OTSS, and recent technologies introduce several unique characteristics. These characteristics moderate the classic requirements change processes. For instance, change in the business processes is a common change driver in many classic or contemporary projects. However, the complexity and structural constraints, as a key characteristic of OTSS, does not allow the organizations to implement all their desired change requests.

These results contribute to the extant requirements literature. Also, we believe they could motivate future studies to examine the influence of other modern technologies (e.g. blockchain) on requirements change. Our fast-changing modern world presents new technologies and digital innovations to us almost every day! The findings of this study also have practical implications. For example, the results suggest that the project managers and business owners include the impact of OTSS and recent technologies on requirements change into their analysis and forecasts. This is crucial, particularly, when they select the more proper technologies to address the business needs of the organization.

As explained in Chapter Four, our *third study* investigated the *joint influence* of change drivers on the changes in requirements in contemporary IS projects. Among the change drivers identified in Chapter Three, two of them were selected in this research — *changes in the environmental factors* and c*hanges in the users' expectations.* To better study the contemporary context, we also included one of the prominent characteristics of OTSS — *complexity and structural constraints* — into our analysis. Therefore, in our research, we examined how these three factors impact the changes in IS requirements. To tackle this query, we employed an agent-based simulation model.

More specifically, we modeled how the technology features in a dynamic environment change, how agents (representing the IS users in real world) change their expectations as a response to the technology changes, and lastly how the organization changes the requirements of the information system based on the new requests of agents. We first developed a base simulation model consisting of these three processes. Next, through two experiments, we explored the influence of two particular constructs on requirements change. These two constructs included: the degree of interdependency in requirements (used to influence the organization's decision) and learning pattern of the population (used to influence the agents' change requests). We also defined two variables to measure the requirements change: information technology value and number of requirements change.

In this study, we inferred interesting results that were all elaborated in detail in Chapter Four. For example, the findings showed that a higher degree of interdependency between requirements can create more challenges for organizations to gain values from the information technologies. We also obtained an interesting finding about the joint influence of the degree of interdependency and the learning pattern of the population on the number of requirements change. Specifically, we explored that a higher degree of interdependency between requirements can reverse the impact of the pattern of learning on the number of requirements changes.

In this research, by using an agent-based simulation method, we add quantitative insights into our previous qualitative outcome. Our study proposes seven theoretical propositions that could provide directions for future research work (e.g., evaluating these propositions). The findings also provide practical implications for IS project managers and human resource managers. Essentially, they elaborate on how managers could create effective interventions, through training and educating the IS users, to increase the value from information technologies.

In sum, by conducting these three studies, we investigated the phenomenon of change in requirements from three different angles. Each of the aforementioned studies has its own limitations that were also discussed in Chapters Two, Three, and Four.

The findings from these studies are starting points for future research. For example, we suggest future case studies examine the efficiency and effectiveness of the Socio-Technical Change Framework and Socio-Technical Requirements Change Method in real-world IS development projects. Also, it would be interesting if future case studies focus on IS projects that utilize packaged solutions and modern technologies (e.g., big data, machine learning, blockchain, etc.). Through that, these case studies could also investigate how the proposed framework and method would be successful in managing IS requirements in the contemporary context.

In addition, we recommend that future research dive more deeply into our findings about the change processes of requirements in modern IS projects. Specifically, it would be important and interesting to elaborate on the unique characteristics of each modern technology that moderate the change processes in IS requirements. We suggest that researchers consider the change mechanisms introduced by the Socio-Technical Change Framework when designing their study. For example, it would be interesting to collect data about the scope of change processes as well as the order, criticality, and frequency of each change driver in a modern IS context. The next step could be mapping the empirical data into the multi-level and socio-technical interactions illustrated by the framework. More critical change drivers and processes could be identified and generalized. The findings would result in developing a mid-range theory.

Another idea would be conducting a longitudinal case study that could reveal the prominent change drivers and moderating impact of modern technologies in different stages of a project. A longitudinal case study could also collect data about the consequence of implementing requirements change in modern IS projects, such as the value of technologies experienced by organizations. Findings from this study could be used by further simulation models to investigate the interdependencies between various change drivers, moderating impacts of modern technologies, and consequences of requirements change.

Moreover, we suggest that future studies synthesize the findings from the first and second studies to design a knowledge-based system. Specifically, this knowledge-based system could integrate various change drivers, change processes, and moderating impacts of OTSS and contemporary technologies into a repository. It could also define different patterns and rules of requirements change explicitly. Based on this knowledge, the system would be able to anticipate possible requirements changes for IS consultants and analysts.

Furthermore, we suggest that future research expand on the scope of requirements change management by integrating the change processes that emerge as a result of the interaction among various business systems within a broader environment. In the modern world, we are facing complex and dynamic environments. In such eco-systems, many heterogeneous business systems are interdependent on each other. This has implications for requirements management research and practice (Berente et al., 2009). Specifically, dynamic changes in the environment and interactions among interconnected business systems should be considered as critical sources of requirements evolution in future studies.

Finally, we hope our research helps both IS scholars and practitioners to enrich their knowledge and expertise on the phenomenon of requirements change in IS projects. We also hope this dissertation inspires future scholars to continue in this research direction.

References

Alter, S. 2006. The work system method: Connecting people, processes, and IT for business results, Larkspur, CA, Work System Press.

Alter, S. 2008. "Defining information systems as work systems: implications for the IS field," *European Journal of Information Systems* (17:5), pp. 448-469.

Alter, S. 2013. "Work system theory: Overview of core concepts, extensions, and challenges for the future," *Journal of the Association for Information Systems* (14:2), pp. 72-121.

Anderson, S. and Felici, M. 2002. "Quantitative aspects of requirements evolution," in *Proceedings of the 26th Annual International* Conference on *Computer Software and Applications Conference,* pp. 27-32.

Appan, R. and Browne, G.J. 2010. "Investigating retrieval-induced forgetting during information requirements determination," *Journal of the Association for Information Systems* (11:5), pp. 250-275.

Appan, R. and Browne, G.J. 2012. "The impact of analyst-induced misinformation on the requirements elicitation process," *MIS Quarterly* (36:1), pp. 85-106.

Benbya, H., Nan, N., Tanriverdi, H. and Yoo, Y. 2020. "Complexity and Information Systems Research in the Emerging Digital World," *Management Information Systems Quarterly* (44:1), pp. 1-17.

Berente, N., Hansen, S. and Lyytinen, K., 2009. "High impact design requirements-key design challenges for the next decade," In *Design Requirements Engineering: A Ten-Year Perspective*. Heidelberg, Springer.

Bergman, M., King, J.L. and Lyytinen, K. 2002. "Large-scale requirements analysis revisited: the need for understanding the political ecology of requirements engineering," *Requirements Engineering* (7:3), pp. 152-171.

Boehm, B. and Basili, V.R., 2001. "Software defect reduction top 10 list," *Computer* (34:1), pp. 135-137.

Bostrom, R.P. and Heinen, J.S. 1977a. "MIS problems and failures: a socio-technical perspective. Part I: the causes," *MIS Quarterly* (1:3), pp. 17-32.

Bostrom, R.P. and Heinen, J.S. 1977b. "MIS problems and failures: a socio-technical perspective. Part II: the application of socio-technical theory," *MIS Quarterly* (1:4), pp. 11-28.

Bowen, P.L., Heales, J. and Vongphakdi, M.T., 2002. "Reliability factors in business software: volatility, requirements and end-users," *Information Systems Journal* (12:3), pp. 185-213.

Browne, G.J. and Rogich, M.B. 2001. "An empirical investigation of user requirements elicitation: Comparing the effectiveness of prompting techniques," *Journal of Management Information Systems* (17:4), pp. 223-249.

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

Carlson, E.D., Grace, B.F. and Sutton, J.A. 1977. "Case studies of end user requirements for interactive problem-solving systems," *MIS Quarterly* (1:1) pp. 51-63.

Chakraborty, S., Sarker, S. and Sarker, S. 2010. "An exploration into the process of requirements elicitation: A grounded approach," *Journal of the association for information systems* (11:4) pp. 212-249.

Cherns, A. 1976. "The principles of sociotechnical design," *Human relations* (29:8), pp. 783-792.

Chung, L., Nixon, B.A. and Yu, E. 1996. "Dealing with change: An approach using non-functional requirements," *Requirements Engineering* (1:4) pp. 238-260.

Dahlstedt, A.G. and Persson, A. 2003. "Requirements interdependencies-moulding the state of research into a research agenda," in *Proceedings of the 9th International Workshop on REFSQ*.

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

Davis, G.B. 1982. "Strategies for information requirements determination," *IBM systems journal* (21:1) pp. 4-30.

Davis, J.P., Eisenhardt, K.M. and Bingham, C.B. 2007. "Developing theory through simulation methods," *Academy of Management Review* (32:2) pp. 480-499.

Ecklund Jr, E.F., Delcambre, L.M. and Freiling, M.J. 1996. "Change cases: use cases that identify future requirements," In *ACM SIGPLAN Notices* (31:10) pp. 342-358.

Gubrium, J.F., Holstein, J.A., Marvasti, A.B. and McKinney, K.D. 2012. The SAGE handbook of interview research: The complexity of the craft, Sage Publications.

Guinan, P.J., Cooprider, J.G. and Faraj, S. 1998. "Enabling software development team performance during requirements definition: A behavioral versus technical approach," *Information systems* research (9:2), pp. 101-125.

Grünbacher, P., Halling, M., Biffl, S., Kitapci, H., and Boehm, B.W. 2004. "Integrating collaborative processes and quality assurance techniques: experiences from requirements negotiation," *Journal of Management Information Systems* (20:4), pp. 10-30.

Hanisch, J. and Corbitt, B. 2007. "Impediments to requirements engineering during global software development," *European Journal of Information Systems* (16:6), pp. 793-805.

Harker, S.D., Eason, K.D. and Dobson, J.E. 1993. "The change and evolution of requirements as a challenge to the practice of software engineering," in *Proceedings of the IEEE International Symposium on Requirements Engineering,* pp. 266-272.

Henver, A.R., March, S. T., Park, J., and Ram, S. 2004. "Design science in information systems research," *MIS Quarterly* (*28*:1), pp. 75-105.

Hickey, A.M. and Davis, A.M. 2004. "A unified model of requirements elicitation," *Journal of Management Information Systems* (20:4), pp. 65-84.

Holmström, J. and Sawyer, S. 2011. "Requirements engineering blinders: exploring information systems developers' black-boxing of the emergent character of requirements," *European Journal of Information Systems* (20:1), pp. 34-47.

Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J. and Robinson, W. 2011. "The brave new world of design requirements," *Information Systems* (36:7), pp. 992-1008.

Jayanth, R., Jacob, V.S. and Radhakrishnan, S. 2011. "Vendor and client interaction for requirements assessment in software development: implications for feedback process," *Information Systems Research* (22:2), pp. 289-305.

Jayatilleke, S., Lai, R. and Reed, K. 2018. "A method of requirements change analysis," *Requirements Engineerin*g (23:4), pp. 493-508.

Kahn, R.L., Wolfe, D.M., Quinn, R.P., Snoek, J.D. and Rosenthal., R.A. 1964. Organizational Stress: Studies in Role Conflict and Ambiguity. Wiley.

Kirsch, L.J. and Haney, M.H. 2006. "Requirements determination for common systems: turning a global vision into a local reality," *The Journal of Strategic Information Systems* (15:2), pp. 79-104.

Kude, T., Mithas, S., Schmidt, C.T. and Heinzl, A. 2019. “How pair programming influences team performance: The role of backup behavior, shared mental models, and task novelty,” *Information Systems Research* (30:4), pp. 1145-1163.

Land, F., 1982. “Adapting to changing user requirements,” *Information & Management* (5:2), pp. 59-75.

Leavitt, H. J. 1965. “Applied organizational change in industry: Structural, technological and humanistic approaches,” In J. G. March (Ed.), *Handbook of organization*. Chicago, Illinois, Rand McNally.

Lee, G. and Xia, W. 2005. “The ability of information systems development project teams to respond to business and technology changes: a study of flexibility measures,” *European Journal of Information Systems* (14:1), pp. 75-92.

Lee, G. and Xia, W. 2010. “Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility,” *MIS Quarterly* (34:1), pp. 87-114.

Levine, S.S. and Prietula, M.J. 2012. “How knowledge transfer impacts performance: A multilevel model of benefits and liabilities,” *Organization Science* (23:6), pp. 1748-1766.

Levinthal, D.A. 1997. “Adaptation on rugged landscapes,” *Management science* (43:7), pp. 934-950.

Lyytinen, K. and Newman, M. 2008. “Explaining information systems change: a punctuated socio-technical change model,” *European Journal of Information Systems* (17:6), pp. 589-613.

Marakas, G.M. and Elam, J.J. 1998. “Semantic structuring in analyst acquisition and representation of facts in requirements analysis,” *Information Systems Research* (9:1), pp. 37-63.

March, J.G. 1991. “Exploration and exploitation in organizational learning,” *Organization Science* (2:1), pp. 71-87.

Marshall, C. and Rossman, G.B. 2016. Designing qualitative research, Sage Publications.

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

Mathiassen, L., Saarinen, T, Tuunanen, T., and Rossi, M. 2007. "A contigency model for requirements development," *Journal of the Association for Information Systems* (8:11), pp. 569-597.

McGee, S. and Greer, D. 2012. "Towards an understanding of the causes and effects of software requirements change: two case studies," *Requirements Engineering* (17:2), pp. 133-155.

Meth, H., Mueller, B., and Maedche, A. 2015. "Designing a requirement mining system," *Journal of the Association for Information Systems* (16:9), pp. 799-837.

Miles, M.B. and Huberman, A.M., 1994. Qualitative data analysis: An expanded sourcebook, Sage Publications.

Moe, C.E., Newman, M. and Sein, M.K. 2017. "The public procurement of information systems: dialectics in requirements specification," *European Journal of Information Systems* (26:2), pp. 143-163.

Mumford, E. 1997. "ETHICS: user led requirements analysis and business process improvement," in *Human Comfort and Security of Information Systems,* pp. 25-38.

Mumford, E. 2000. "A socio-technical approach to systems design," *Requirements Engineering* (5:2), pp. 125-133.

Mumford, E. 2006. "The story of socio-technical design: Reflections on its successes, failures and potential," *Information systems journal* (16:4), pp. 317-342.

Mumford, E. and Weir, M. 1979. Computer systems in work design--the ETHICS method: effective technical and human implementation of computer systems: a work design exercise book for individuals and groups. John Wiley & Sons

Munro, M.C. and Davis, G.B. 1977. "Determining management information needs: a comparison of methods," *MIS Quarterly* (1:2), pp. 55-67.

Nan, N. and Tanriverdi, H. 2017. "Unifying the Role of IT in Hyperturbulence and Competitive Advantage Via a Multilevel Perspective of IS Strategy," *MIS Quarterly* (41:3), pp. 937-958.

Nurmuliani, N., Zowghi, D. and Powell, S. 2004. "Analysis of requirements volatility during software development life cycle," in *Proceedings of the Software Engineering Conference,* pp. 28-37.

Patel, N.V., 1999. "The spiral of change model for coping with changing and ongoing requirements," *Requirements Engineering* (4:2), pp. 77-84.

Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. 2007. "A design science research methodology for information systems research," *Journal of management information systems* (24:3), pp. 45-77.

Pitts, M.G. and Browne, G.J. 2004. "Stopping behavior of systems analysts during information requirements elicitation," *Journal of management information systems* (21:1), pp. 203-226.

Pitts, M.G. and Browne, G.J. 2007. "Improving requirements elicitation: an empirical investigation of procedural prompts," *Information systems journal* (17:1), pp. 89-110.

Polkinghorne, D.E. 2005. "Language and meaning: Data collection in qualitative research," *Journal of Counseling Psychology* (52:2), pp. 137-145.

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

Ramesh, B., Mohan, K. and Cao, L. 2012. "Ambidexterity in agile distributed development: an empirical investigation," *Information systems research* (23:2), pp. 323-339.

Ramesh, B., Cao, L. and Baskerville, R. 2010. "Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal* (20:5), pp. 449-480.

Rolland, C., Salinesi, C. and Etien, A. 2004. "Eliciting gaps in requirements change," *Requirements Engineering* (9:1), pp. 1-15.

Rosenkranz, C., Charaf, M.C. and Holten, R. 2013. "Language quality in requirements development: tracing communication in the process of information systems development," *Journal of Information Technology* (28:3), pp. 198-223.

Rosenkranz, C., Vranesic, H. and Holten, R. 2014. "Boundary interactions and motors of change in requirements elicitation: A dynamic perspective on knowledge sharing," *Journal of the Association for Information Systems* (15:6), pp. 306-345.

Sarnikar, S., El-Gayer, O., Wahbeh, A. 2014. "A process model for eliciting requirements of socio-technical systems," in *Proceedings of the 20th Americas Conference on Information Systems.*

Schneider, S., Wollersheim, J., Krcmar, H. and Sunyaev, A. 2018. “How do requirements evolve over time? A case study investigating the role of context and experiences in the evolution of enterprise software requirements,” *Journal of Information Technology* (33:2), pp. 151-170.

Schultze, U. and Avital, M. 2011. “Designing interviews to generate rich data for information systems research,” *Information and organization* (21:1), pp. 1-16.

Shuraida, S. and Barki, H. 2013. “The influence of analyst communication in IS projects,” *Journal of the Association for Information Systems* (14:9), pp. 482-520.

Sutton, D.C. 2000. “Linguistic problems with requirements and knowledge elicitation,” *Requirements Engineering* (5:2), pp. 114-124.

Taber, C.S. and Timpone, R.J. 1996. Computational modeling, Sage Publications.

Thanasankit, T. 2002. “Requirements engineering—exploring the influence of power and Thai values,” *European Journal of Information Systems* (11:2), pp. 128-141.

Tuunanen, T. and Kuo, I.T. 2014. “The effect of culture on requirements: a value-based view of prioritization,” *European Journal of Information Systems* (24:3), pp. 295-313.

Urquhart, C. 2001. “Analysts and clients in organisational contexts: a conversational perspective,” *The Journal of Strategic Information Systems* (10:3), pp. 243-262.

Venable, J., Pries-Heje, J. and Baskerville, R. 2016. “FEDS: a framework for evaluation in design science research,” *European journal of information systems* (25:1), pp. 77-89.

Vitharana, P., Zahedi, F. and Jain, H.K. 2016. “Enhancing analysts’ mental models for improving requirements elicitation: a two-stage theoretical framework and empirical results,” *Journal of the Association for Information Systems* (17:12), pp. 804-840.

Woo, C., Saghafi, A. and Rosales, A. 2014. “What is a Contribution to IS Design Science Knowledge?,” in *Proceedings of the 35th International Conference on Information Systems*.

Xiao, X., Lindberg, A., Hansen, S., and Lyytinen, K. 2018. "Computing" requirements for open source software: A distributed cognitive approach,” *Journal of the Association for Information Systems* (19:12), pp. 1217-1252.

Yu, E. 2011. Modelling strategic relationships for process reengineering. In *Social Modeling for Requirements Engineering*. MIT Press, pp. 11-152.

Yu, E. 1997. "Towards modelling and reasoning support for early-phase requirements engineering," in *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, pp. 226-235.

Zappavigna, M. and Patrick, J. 2010. "Eliciting tacit knowledge about requirement analysis with a Grammar-targeted Interview Method (GIM)," *European Journal of Information Systems* (19:1), pp. 49-59.

Zhang, H., Li, J., Zhu, L., Jeffery, R., Liu, Y., Wang, Q., and Li, M. 2014. "Investigating dependencies in software requirements for change propagation analysis," *Information and Software Technology* (56:1), pp. 40-53.

Zowghi, D. and Gervasi, V. 2004. "Erratum to "On the interplay between consistency, completeness, and correctness in requirements evolution," *Information and Software technology* (46:11), pp. 763-779.

Zowghi, D. and Nurmuliani, N. 2002. "A study of the impact of requirements volatility on software project performance," in *Proceedings of the Ninth Asia-Pacific Software Engineering Conference,* pp. 3-11.

Zowghi, D. and Offen, R. 1997. "A logical framework for modeling and reasoning about the evolution of requirements," in *Proceedings of the Third IEEE International Symposium on Requirements Engineering,* pp. 247-257.

Appendices

Appendix A: Invitation Letter to Managers

Dear Sir/Madam,

I am writing this letter regarding a research study on ***changes in information systems requirements***. This study is conducted under the supervision of **Professor Carson Woo** in Accounting and Information Systems Division at Sauder School of Business at UBC.

I would appreciate it if you could pass on this message and our invitation letter to your business analysts and system analysts. I would like to invite them to participate in an interview about their experience of changes in requirements. Their responses would help us understand the drivers of change and the evolution process in information systems requirements. Based on this information, we can then propose a method that can help business analysts and system analysts anticipate some potential changes before they actually happen. We expect this will mitigate the risks and costs associated with later change implementations.

I would appreciate it if your business analysts and system analysts who receive my invitation could send me a message in the next 5 days if they are interested in participation. I will then send them additional information regarding the time and format of the interview, as well as a consent form for their participation. The confidentiality of their responses will be respected. We appreciate your and their attention to this request.

Please do not hesitate to let me know if you have any further questions.

Sincerely,
Atefeh Taghavi
PhD Student in Accounting and Information Systems Division
Sauder School of Business
University of British Columbia
Email: atefeh.taghavi@sauder.ubc.ca

Appendix B: Invitation Letter to Analysts and Consultants

Dear Madam/Sir,

I am writing this letter regarding a research study on ***changes in information systems requirements***. This study is conducted under the supervision of **Professor Carson Woo** in Accounting and Information Systems Division at Sauder School of Business at UBC.

I would like to invite you to participate in an interview about your experience of changes in information systems requirements. Your responses would help us understand the **reasons behind the changes** in requirements.

- **Expected results of the study:**

Based on the information collected from our interviews, we are going to develop a model of the change processes in requirements. This model will provide business analysts, system analysts, consultants, and project managers with a set of guidelines to more efficiently manage the requirements in their future projects, for instance by anticipating some potential requirements changes before they actually happen.

- **How can our research contribute to your profession?**

As business analysts or consultants we all may have experienced change requests from our clients that we wished we could have some ideas about at the beginning of the project. In our research, we aim to collect information from various change requests before or after implementing a system related to any of the recent technologies mentioned below*. If you are interested to learn more about the change processes in requirements, we would be happy to share our research findings with you later. You just need to give us your contact information after the interview process. We hope our research results can be added to your current body of knowledge, and consequently assist you to more efficiently dealing with changing requests and proactively considering their possibility before they actually emerge.

- **Privacy concerns:**

We should mention that the data we collect from your previous experiences is solely for research and academic purposes. We do not think there is anything in this study that could harm you or your organization. We will not collect any sensitive personal information. Also, we will not share your opinions with your organization. Moreover, you, your firm, and your projects will not be identified by name in any reports of the study.

- **Participation:**

If you are interested to participate in our interview process and your background meets the following criteria, please reply to this invitation. We will then send you more details regarding the format of the interview. Each interview will take approximately 45-60 minutes. Your participation is voluntary, and you can terminate the interview anytime if you decide to do so without any ramifications. For the interview session, we will find a location that is convenient to you.

***Interviewee criteria:**
Work experience in a requirements elicitation or system analysis project related to any of the following modern technological capabilities: ***off the shelf systems, cloud computing, mobility***, ***virtual teams***, ***e-commerce***, ***artificial intelligence (or algorithm-based software)***, ***big data***, ***machine learning***, ***blockchain***, ***or internet of things***.

If you have any questions, please do not hesitate to contact me.

Thank you for your support and looking forward to hearing from you.

Sincerely,
Atefeh Taghavi
PhD Student in Accounting and Information Systems Division
Sauder School of Business
University of British Columbia
Email: atefeh.taghavi@sauder.ubc.ca

Appendix C: Interview Questions

Background information:

1. For how long have you been doing business analysis (or system analysis)?
   1.1. In which type of firms have you been working? (e.g., in-house projects or consulting services to other firms, as well as the nature, size, and age of your firm)
   1.2. Did you have previous related positions that might help in your BA experience?

Change Requests Experience:

As you know, we are collecting information about the requirements changes in the off-the-shelf systems, related to any modern technological feature mentioned in our invitation email.

2. In which of those areas have you been working? (For example, *cloud computing, mobility*, *virtual teams*, *e-commerce*, *artificial intelligence (or algorithm-based software)*, *big data*, *machine learning*, *blockchain*, *or internet of things*)
3. Have you experienced any project in which your clients asked for changes in their requirements, either during or after implementing the system?
   3.1. Can you tell me more about some of those cases?
      3.1.1. Please feel free to think about your last 2-3 cases.
   3.2. How common are those change requests in your work?

Change drivers and roots:

4. What had happened that your clients needed the changes? What were some of the causes of those change requests?
   4.1. [Unplanned prob. questions to better understand "why" and "how" the change request was formed]

Response to change requests:

5. How did your team respond the change requests? For example, did you accept to address the new requests?
   5.1. *[Unplanned prob. questions to understand how the request was responded]*

Anticipation of changes:

6. Now that you look at those cases, do you think if it was possible that your client or your development team could anticipate/tackle the change requests before they emerged?
   6.1. For instance, do you think if there was any sign for a change from the beginning? Can you give me an example of that?

Wrap up questions:

7. If you have also experienced the change requests in other types of system development (other than off-the-shelf packages), such as designing a custom software, do you see any differences between the change requests in them? How about the modern technological capabilities vs. traditional classic features?
8. Is there anything else that you would like to add that was not covered in our conversation?

Appendix D: Invitation Letter to Managers (for the Survey Phase)

Dear Sir/Madam,

I am writing this letter regarding a research study on changes in information systems requirements. This study is conducted under the supervision of Prof. Carson Woo in Accounting and Information Systems Division at Sauder School of Business at UBC.

This research aims to investigate the change processes in information systems requirements, particularly in our modern technology era. In this study, we would like to invite expert system analysts, business analysts, and business technology consultants to participate in an online survey and share their experience of change processes in requirements with us. I would greatly appreciate it if you could please pass on this invitation to the expert system analysts, business analysts and business technology consultants in your team. Their responses would help us tremendously understand the drivers of change and the change process in information systems requirements. Based on this information, we can then propose a framework that can help business analysts and system analysts to deeply understand the roots of changes in requirements and anticipate some of those changes before they happen. We expect this outcome will mitigate the risks and costs associated with later change requests and implementation.

I would truly appreciate it if the business analysts, system analysts, and business technology consultants in your team who receive my invitation and are interested to participate in our survey could please send me a message in the next few days. I will then send them more detailed information regarding the survey process, as well as a consent form for their participation. We will not ask any personal or confidential questions about the details of their project. We will focus on the general patterns of changes that can happen in most of the information systems projects. The confidentiality of their responses will be respected. We greatly appreciate your and their attention to this request.

Please do not hesitate to let me know if you have any further questions.

Sincerely,
Atefeh Taghavi
PhD Student in Accounting and Information Systems Division
Sauder School of Business
University of British Columbia
Email: atefeh.taghavi@sauder.ubc.ca

Appendix E: Survey Questions

Background information:

1) For how long have you been doing system analysis, business analysis, or consulting?
2) In which domains or sectors have you worked as a business analyst, system analyst or consultant? For example, finance, accounting, higher education, healthcare, government, etc.
3) In which type of projects have you worked (for example, working at a consulting firm, in-house team, or both)?

Main section:

We have already collected 14 potential roots of cause of change in information systems requirements. We would appreciate it if you could please tell us whether you have experienced any of these 14 change reasons or not. Even if you have not directly experienced them, as long as you think they are considered as the reasons for requirements change, please let us know.

We will then ask you whether you think modern system development solutions (such as using off-the-shelf packages and cloud systems) or modern technologies (such as mobile systems, e-commerce, artificial intelligence/machine learning, big data, or any other modern technologies you have worked with) have an impact on the way each of those 14 change reasons lead to actual changes.

For instance, you might think, by using modern system development solutions, change driver X generates more/or fewer changes in requirements, compared with situations when we use classic system development practices (such as designing and building a system from the scratch).

Regarding modern technologies, please let us know if you think when designing systems related to modern technologies (e.g., systems based on mobile systems, e-commerce, artificial intelligence/machine learning, big data,...) the changes in requirements might be different.

Please remember that it is perfectly fine if you do not observe any difference in requirements change when dealing with modern system development solutions or modern technologies.

List of 14 change drivers:

1. Lack of clients involvement
2. Oversight of details and recalling issue by both clients and analysts

3. Lack of clear communication between clients and analysts
4. Lack of transparency from analysts
5. Lack of a proper understanding of the project by both clients and analysts
6. Lack of a clear understating of the business processes in the organization by clients
7. Lack of considering the technical standards by analysts
8. Changes in the technical knowledge of clients
9. Changes in the clients' experience
10. Changes in the clients' expectations
11. Changes in the connected systems
12. Changes in the project
13. Changes in organizational factors
14. Changes in factors from outside the organization

Main questions:

[Name of the change driver and a brief description will be show up.]

1) Do you have the same experience or do you agree with this change reason?
   □ Yes □ No

[If a participant's answer is yes, then the following questions (2-8) will show up. Otherwise, s/he will see the first question for the next change driver.]

2) Can you please identify how often this factor has led to changes in requirements in your previous projects?
   □ I have not faced this change reason so far.
   □ I have faced this change reason only in a few cases.
   □ I have faced this factor as a dominant change reason in the majority of my projects

3) Do you think the mentioned change process could be different when we implement "off-the-shelf systems" compared to "designing a system from the scratch"?
   □ Yes □ No

[If a participant's answer is yes, then questions 4-5 will show up.]

4) If you chose "yes" in question 3, please specify the type of "off-the-shelf systems" you referred to.
   □ Cloud-based
   □ On-premise (purchased and installed locally on clients' computers)

5) If you chose "yes" in question 3, can you please explain the difference?

6) Do you think the mentioned change process could be different when clients dealing with any of the “modern technologies” (e.g., mobile systems, e-commerce, artificial intelligence-based and machine learning, big data, etc.)?
   □ Yes □ No

[If a participant’s answer is yes, then questions 7-8 will show up.]

7) If you chose “yes” in question 6, please specify the type of “modern technologies” you are referring to:
   □ Mobile systems □ E-commerce □ AI-based and machine learning
   □ Big data □ Others
   If you chose others, please name the technology: ------------

8) If you chose “yes” in question 6, can you please explain the difference?

[The same set of questions is repeated for each of the 14 change drivers]

Closing questions:

1) Besides the change reasons we discussed in this survey, is there any other change drivers that you have experienced or you are aware of that you would like to share with us? If so, would you also please let us know how you see the impact of modern system development solutions or modern technologies on those change drivers?

2) If you have any feedback on the structure of our survey (for example, questions, timing, etc.) we would greatly appreciate it you could please let us know.

Appendix F: Pseudo-Code of the Simulation Model

Each requirement owns [
requirement-ID
requirement-value
]

Each user owns [
user-ID
requirement-set
learning-rate
\\ learning-rate high = 1, learning-rate medium = 2, and learning-rate low = 3
proposed-technology-value
]

Set requirements number = **20**
\\ This shows the number of requirements in the organizational information system.

Set users number = **40**
\\ This shows the number of users in the organization.

INITIAL SETUP of the SIMULATION {
Setup the technology features pool
Setup information system requirements
Setup agents
\\ For other experimental conditions, call **Setup agents-second-experimental-condition** or **Setup agents-third-experimental-condition** instead of **Setup agents**.
Setup the environment
}

\\ The following are the procedures called by "INITIAL SETUP of the SIMULATION".

Setup the technology features pool {
\\ This procedure creates a pool of technology features (s = 40). At each iteration, available technologies on the market are randomly chosen from this pool.
Read the number of technology features (s) and value of each technology feature from a file
Create s technology features
Assign the ID and value of each technology feature accordingly
\\ ID is a number between 0 and s.
}

Setup information system requirements {
\\ This procedure sets the initial requirements of the information system.

information system requirements = **Create** a set of requirements with the length of **requirements number**
For each requirement:
    Set requirement-ID = a random number between 0 and s
    Set requirement-value
    \\ By using the requirement-ID, this command finds the value of each requirement from the technology features pool.
Calculate technology-value of the **information system requirements**
\\ This is the sum of requirement-value of requirements in the set.
}

Setup agents {
\\ This procedure creates the agents and assign value to their attributes. The initial requirement-set of agents is the same as the requirements of the organizational information system.
Create 40 users
\\ (users number = 40)
For each user:
    Set user-ID = a random number between 0 and 40
    Set requirement-set = **information system requirements**
    Set learning-rate = a random number between 1 and 3
    Set proposed-technology-value = **technology-value**
}

Setup agents-second-experimental-condition {
\\ This procedure is similar to the original set-up agents procedure, but the distribution of learning rates is different here.
Create 40 users
\\ (users number = 40)
For each user:
    Set user-ID = a random number between 0 and 40
    Set requirement-set = **information system requirements**
    Set proposed-technology-value = **technology-value**
For 1/3 of users:
    Set learning-rate = a random number between 2 and 3
For 2/3 of users:
    Set learning-rate = 1
}

Setup agents-third-experimental-condition {
\\ This procedure is similar to the original set-up agents procedure, but the distribution of learning rates is different here.
Create 40 users
\\ (users number = 40)
For each user:
    Set user-ID = a random number between 0 and 40

Set requirement-set = **information system requirements**
Set proposed-technology-value = **technology-value**
For 1/3 of users:
Set learning-rate = a random number between 1 and 2
For 2/3 of users:
Set learning-rate = 3
}

Setup the environment {
modern technology features = **Create** a set of requirements with the length of **requirements number**
\\ This set shows the available technologies on the market.
For each requirement:
Set requirement-ID = a random number between 0 and s
}

\\ End of INITIAL SETUP procedures

RUN {
\\ This procedure is called 50 times continuously in our simulation.
IF (**ticks** < 50) [
Users learn modern tech
Organization learns from users with K = 0
\\ Depending on the experimental condition, call **Organization learns from users with K = 4** instead.
Update the environment
ticks = ticks + 1
]
Set Average-of-number-of-change = **Sigma-number-of-change** / (**ticks**)
Set Average-of-technology-value = **Sigma-technology-value** / (**ticks**)
\\ These variables are calculated when the organization learns from users.
}

\\ The following are the procedures called by "RUN".

Users learn modern tech {
\\ This procedure asks users to update their requirements-set based on technologies available on the market.
Ask users [
IF **learning-rate** = 1 [
Let Counter = **Requirements number** /2
Repeat Counter [
Randomly choose one of the requirements in **requirement-set**
Let pointer = position of this requirement in **requirement-set**
Replace requirement-set (**pointer**) = **modern technology features** (**pointer**)

```
            ]
        ]
    \\ Maximum 10 proposed requirements of these users can be changed at each iteration.
    IF learning-rate = 2 [
        Let Counter = Requirements number /3
          Repeat Counter [
              Randomly choose one of the requirements in requirement-set
              Let pointer = position of this requirement in requirement-set
              Replace requirement-set (pointer) = modern technology features
             (pointer)
              ]
        ]
    \\ Maximum 6 proposed requirements of these users can be changed at each iteration.
    IF learning-rate = 3 [
        Let Counter = Requirements number /5
          Repeat Counter [
              Randomly choose one of the requirements in requirement-set
              Let pointer = position of this requirement in requirement-set
              Replace requirement-set (pointer) = modern technology features
             (pointer)
              ]
        ]
    \\ Maximum 4 proposed requirements of these users can be changed at each iteration.
  ]
For each user:
    Calculate proposed-technology-value
\\ This is the sum of requirement-value of requirements in the set.
}
```

Organization learns from users with K = 0 {

\\ This procedure compares the proposed-technology-value of all agents with the current technology-value of the organizational information system. An agent whose proposed-technology-value is greater than the technology-value is chosen. The information system requirements are updated based on the proposed requirement-set of the winner agent.

```
Ask users [
    IF (technology-value  < proposed-technology-value) [
        Set dummy-array = requirement-set
        Set dummy-tech-value = proposed-technology-value
    ]
Calculate number of change (dummy-array)
Set information system requirements = dummy-array
Set technology-value = dummy-tech-value
Set Sigma-technology-value = Sigma-technology-value + technology-value
```

\\ The initial value of Sigma-technology-value is 0. This variable is used to calculate the average-of-technology-value.
}

Calculate number of change (**dummy-array**) {
\\ This procedure calculates the number of changes in information system requirements.
Set number-of-change = 0
Set Pointer = 0
Repeat requirements number [
    IF (**information system requirements** (**Pointer**) **!= dummy-array** (**Pointer**))
        Set number-of-change = number-of-change + 1
    Set Pointer = **Pointer** + 1
    ]
Set Sigma-number-of-change = Sigma-number-of-change + number-of-change
\\ The initial value of Sigma-number-of-change is 0. This variable is used to calculate the average-of-number-of-change.
}

Organization learns from users with K = 4 {
\\ This procedure is similar to the original organizational learning procedure (Organization learns from users with K = 0). However, this one considers the interdependencies between requirements.

Ask users [
    IF (**technology-value** < **proposed-technology-value**) [
        Set dummy-array = **requirement-set**
        Set dummy-tech-value = **proposed-technology-value**
    ]
dummy-array = **Apply dependencies** (**dummy-array**)
Calculate number of change (**dummy-array**)
Set information system requirements = **dummy-array**
Calculate technology-value
\\ This is the sum of requirement-value of requirements in the information system.
Set Sigma-technology-value = Sigma-technology-value + technology-value
}

Apply dependencies (**dummy-array**) {
\\ This procedure applies the interdependency constraints between requirements (K = 4).
Set final-array = **information system requirements**
\\ Initial value of the final list of requirements
Set array-pointer = 0
Repeat requirements number [
    \\ Accept the last 4 requirements proposed by the user without any change.
    IF **array-pointer** refers to one of the last 4 requirements in the list
        Replace final-array (**array-pointer**) = **dummy-array** (**array-pointer**)
    \\ Accept the requirements with an odd requirement-ID.

```
        ELSE IF dummy-array (array-pointer) has an odd requirement-ID
                Replace final-array (array-pointer) = dummy-array (array-pointer)
        \\ For requirements with an even requirement-ID, accept them only if they meet the
        following conditions.
        ELSE IF dummy-array (array-pointer) has an even requirement-ID [
                IF all 4 neighbors have odd requirement-ID
                   Replace final-array (array-pointer) = dummy-array (array-pointer)
                ]
        Set array-pointer = array-pointer + 1
        ] \\ End of loop
Return final-array
}
```