HEC MONTRÉAL

École affiliée à l'Université de Montréal

A Decision Support System for Software Project Risk Management

A Three-Essay Dissertation

par

Mazen El-Masri

Thèse présentée en vue de l'obtention du grade de Ph. D en administration, option technologies de l'information

Juillet, 2013

HEC MONTRÉAL

École affiliée à l'Université de Montréal

Cette thèse intitulée

A Decision Support System for Software Project Risk Management

A Three-Essay Dissertation

Présenté par:

Mazen El-Masri

a été évaluée par un jury composé des personnes suivantes :

Jacques Robert
HEC Montréal
Président-rapporteur

Suzanne Rivard
HEC Montréal
Présidente du comité de surveillance

Henri Barki
HEC Montréal
Membre du jury

Yves Pigneur
HEC Lausanne
Examinateur externe

Résumé

Dans la recherche sur les systèmes d'information (SI), les risques liés aux projets de développement de systèmes sont souvent décrits comme un obstacle central à la réussite du projet. Les progrès importants en recherche SI comprennent l'identification des facteurs de risque inhibant la réussite des projets, ainsi que des pratiques de gestion qui en restreignent les effets. Récemment, des études exploratoires ont révélé des risques plus complexes que ce qui était estimé auparavant à cause des interactions dynamiques de leurs composantes. Pourtant, il n'y a pas de cadre théorique expliquant ces interactions. De plus, les outils logiciels dont disposent les praticiens pour identifier et gérer les risques des projets SI ne sont pas ajustés à ces particularités.

Par conséquent, cette thèse adopte une approche relevant de la science du design afin de définir les principes de conception d'un système d'aide à la gestion du risque des projets SI qui suive les interactions dynamiques des composantes du risque. Deux construits sont ici essentiels : le risque de projet SI et sa réussite. La littérature SI offre cependant diverses conceptualisations de ces deux construits, ce qui mérite d'être reconsidéré. Ainsi, le premier essai visait à proposer une conceptualisation claire du construit de réussite de projet. La littérature publiée entre 1990 et 2012, portant sur la réussite de projets a été analysée selon deux dimensions : les processus constituant un projet SI et les parties prenantes au projet. Notre analyse a résulté en un construit constitué de l'arrimage de la réussite de chacun de trois processus centraux (développement, mise en œuvre, et assimilation) et de leurs conséquences (de production, d'organisation, et d'affaires) telles que perçues par les différentes parties prenantes.

Notre second essai offre une nouvelle conceptualisation du construit du risque de projet SI. Pour découvrir les interactions entre les composantes du risque, il fallait recourir à une conceptualisation compréhensive qui distingue les dimensions du

construit et leurs façons de s'inter-relier. Nous avons donc recensé la littérature de risque de projet SI sur une période de 23 ans (1990-2012) pour développer un construit du risque intégrant toutes les conceptions existantes. Une analyse sémantique des définitions du risque et d'autres termes descriptifs de la littérature SI a été faite. Cette analyse nous a permis de distinguer quatre dimensions – attributs (traits) du projet, événements indésirables, pratiques de gestion du risque, et résultats visés – et de comprendre en quoi elles sont inter-reliées. Ainsi, nous avons conçu le risque comme un risque résiduel, défini comme la probabilité d'une déviation d'un résultat visé conditionnelle à l'adoption de pratiques de gestion de risque. Afin de tester les relations entre les composantes du construit tel que conceptualisé, nous avons développé et évalué un modèle causal qui correspond à la structure interne du construit et aux relations de ses composantes. Nous avons testé le modèle en utilisant les données de 82 cas – publiés – portant sur la gestion de risque de projets SI. Le schéma d'encodage a été validé auprès d'experts SI. Les données obtenues ont été analysées avec PLS et les résultats soutiennent le modèle.

Le troisième essai, s'appuie sur la conceptualisation proposée dans le second essai pour spécifier les principes de conception d'un système de gestion du risque des projets SI qui tienne compte des interactions dynamiques des différentes composantes de risque. Afin de parvenir à détecter ces interactions, le construit de risque décrit dans le second essai a été raffiné grâce à la subdivision de ses dimensions en sous-composantes. Les données de cas existants ont été utilisées pour déterminer les patterns d'interrelation entre les composantes du risque, et ensuite les interpréter pour en déduire des principes de conception d'artéfacts TI. Nous avons utilisé un logiciel de simulation pour développer un prototype d'artéfact TI qui exprime les principes du concept proposé. Pour évaluer la validité et l'utilité de cet artéfact TI, nous l'avons utilisé dans une intervention en gestion du risque de la mise en œuvre d'un projet SI dans un contexte organisationnel réel. L'artéfact TI a aussi été validé par la simulation de scénarios de risque pour trois cas de projets SI et la comparaison de ses résultats avec la réalité décrite dans ces cas. Les résultats obtenus montrent qu'un artéfact TI conforme à nos principes de conception fournit un soutien réel à l'analyse du risque et à la prise de décision.

Mots-clés: Le risque résiduel de projet logiciel, gestion des risques de projet logiciel, la réussite du projet de logiciel, science de la conception, étude de cas, simulation.

Abstract

In the information systems (IS) literature, software project risk is frequently described as a key detriment to project success. Important advances in IS research include the identification of risk factors that are negatively associated with project success as well as management practices that mitigate their effects. More recently, exploratory studies revealed that risk is more intricate than previously thought due to the dynamic interactions between risk components. Yet, there is no theoretical framework to explain such interactions. Moreover, software that practitioners use to help identify and manage risk is not tailored to address such particularities of software project risk.

Accordingly, this dissertation takes a design science approach to articulate the design principles of a software project risk management system that accounts for the dynamic interrelations between risk components. Two constructs – software project risk and software project success – are essential to our study. However, the IS literature presents diverse conceptualizations of both constructs which merits their re-examination. Accordingly, the project success construct was reconceptualized and specified in the first article. Both the process and stakeholder's perspectives were adopted to synthesize 23 years of relevant IS literature (1990-2012). Our analysis resulted in a success construct specified as the combination of the successes of three core processes (development, implementation, and assimilation) and their associated outcomes (product, organizational, and business) as perceived by the different project stakeholder groups.

The software project risk construct was specified in the second article. A comprehensive conceptualization that distinguishes the construct's dimensions and how they interrelate was required to allow for the discovery of interactions between risk components. Therefore, we made use of 23 years of software project risk literature (1990- 2012) to develop a risk construct that integrates the extant

conceptualizations. Semantic analysis was performed on existing risk definitions and other descriptive texts from the IS literature. This analysis allowed us to distinguish four risk dimensions – project traits, undesirable events, risk management practices, and expected outcomes – and understand how they relate to one another. Accordingly, we conceptualized risk as residual risk and define it as the conditional probability of deviation from expected outcomes given that available management practices are considered. To provide support for the proposed residual risk construct, we developed and evaluated a causal model that reflects the construct's internal structure and the relationships between its components. To this end, a case survey of 82 software project cases was conducted. The coding scheme was specified with a panel of experts who helped reclassify existing project risk operationalizations according to the dimensions of the proposed construct. The resulting dataset was analyzed with PLS providing support for the construct's structure.

In the third article, we use the findings from the first two articles to specify the design principles of a software project risk management system that accounts for the dynamic interactions among the different components of risk. To be able to detect the components' interactions, the risk construct specified in the second article was refined further by dissecting its dimensions into subcomponents. A case survey was conducted to observe patterns of interplay among risk components which were then interpreted to derive the IT artifact's design principles. We used simulation software to develop a prototype of an IT-artifact that expresses the proposed design principles. To assess the validity and utility of the IT artifact, it was used in an intervention in the risk management of a software implementation project in a real organizational setting. The IT artifact was also validated by simulating risk scenarios of three software project cases and contrasting its results with the realities described in the cases. The results show that an IT artifact that conforms to the proposed design principles provided actual risk analysis and decision support.

Keywords: Software project residual risk, software project risk management, software project success, case survey, simulation, design science.

Table of Contents

Résumé iii

Abstract vi

Table of Contents ix

List of Tables xiv

List of Figures xv

Acknowledgements xvi

Chapter I – Introduction 1

- Research Objective 2
- Research Question 3
- Research Methodology 3
- Motivation And Contribution 6
- Paper 1: Specifying Software Project Success 7
  - *Research Objective* *8*
  - *Research Method* *8*
  - *Conceptual Model* *8*
  - *Contribution* *9*
- Paper 2: Specifying Software Project Residual Risk 9
  - *Research Objective* *10*
  - *Research Method* *10*
  - *Conceptual Model* *10*
  - *Contribution* *11*
- Paper 3: Towards a Design Theory for Software Project Risk Management Systems 11
  - *Research Objective* *12*
  - *Research Method* *12*
  - *Contribution* *13*
- References 14

Chapter II – Article 1: Specifying Software Project Success 17

Abstract 18

Introduction..19

Identifying The Relevant Literature..22

Research Method 24

Conceptualizations of Software Project Success 25

A Process Perspective of Software Project Success 29

Identifying the Core Processes of the Software Project Lifecycle 29

Organizing the Existing Success Criteria 32

A Stakeholder Perspective of Software Project Success...37

Organizing the Existing Success Criteria 38

Discussion...42

Conclusion 44

References...45

Appendix A: Key Articles that Address the Notion of Software Project Success..................55

Chapter III – Article 2: Specifying Software Project Risk 56

Abstract 57

Introduction...58

Software Project Risk: The Need for Conceptual Clarification......................................60

Risk as Risk Factors 61

Risk as Risk Exposure 62

Risk as Variance of Outcomes.. 62

Risk Factors and the Software Project Risk Construct 63

Reconceptualizing Software Project Risk...64

Components of Software Project Risk.. 66

Internal Structure of the Software Project Risk Construct 71

Providing Evidence for the Relationships between Components of the Software Project Residual Risk Construct...77

Hypothesis Development.. 77

Specifying the Measurement Model 79
Data Collection Methodology 82
Case Selection Criteria *83*
Inclusion Criteria *85*
Exclusion Criteria *85*
Sample Size *86*
Coding Scheme *86*
Data Analysis 87
Evaluation of the Software Project Residual Risk Model 88
Discussion 88
Limitations 92
Conclusion 93
References 95
Appendix A: Definitions of Software Project Risk 100
Appendix B: Definitions of Software Project Risk Factors 103
Appendix C: Selected Journals 106
Appendix D: List of Articles that Identified Risk Factors 109
Appendix E: Card Sorting – Part 1 115
Appendix F: Articles that Identified Risk Management Practices 123
Appendix G: Description of the Semantic Decompositional Analysis that was Conducted. 125
Appendix H: Description of the Card Sorting Exercise – Part 2 126
Appendix I: Existing Categories of Risk Management Practices 130
Appendix J: Case Survey Coding Process 133
Appendix K: Articles Included in the Case Survey 136
Appendices References 144
Chapter IV – Article 3: Design Principles for Software Project Risk Management Systems 147

Abstract..148

Introduction..149

Theoretical Framework..151

Methodology..153

Theory Development..155

Refining the Software Project Residual Risk Construct.. 155

Identifying Patterns of Interactions among Residual Risk Components.......................... 158

Pattern #1: Multiplicative Effect of Project Traits *161*

Pattern #2: Sequential Effect of Undesirable Events.. *163*

Pattern #3: The Presence of a Third Variable.. *165*

Pattern #4: The Tradeoff When Selecting Risk Management Practices *169*

Design Principles for Software Project Risk Management Systems............................... 170

The Association Principle *171*

The Regulation Principle *173*

The Simulation Principle.. *173*

The Self-Learning Principle.. *174*

Theory Demonstration..174

Determining the Strength of Interrelationships between Components of Residual Risk . 175

Producing An Instantiation of the IT Artifact *181*

Theory Evaluation..182

Intervention in the Management of Risk of a Software Implementation Project 182

Assessing Artifact Validity.. 189

Discussion..192

Limitations..195

Conclusion..197

References..200

Appendix A: The Components of Residual Risk Dimensions..220

Appendix B: Case Survey Search Criteria and the Selected Articles 230

Appendix C: Project Risk Assessment Questionnaire 239

Appendix D: Project Risk Evaluation Report 252

List of Tables

Table 2.1 Number of Articles Identified by Journal 23
Table 2.2 Organization of the Identified Literature 25
Table 2.3 The Nature of Success Criteria of the Second Group of Articles 26
Table 2.4 Conceptualizations of Process Success 27
Table 2.5 Conceptualizations of Outcome Success 28
Table 2.6 Classifying Software Project Success Criteria Using the Proposed Model 33
Table 2.7 A Multi-dimensional Multi-stakeholder Software Project Success Construct 41
Table 4.1 Inter-rater agreement of the project traits and undesirable events components 157
Table 4.2 Sample of the Scenarios on the Multiplicative Effect of Project Traits 162
Table 4.3 Sample of the Probability, Impact, and Risk Exposure from Han and Huang 164
Table 4.4 Sample of the Scenarios Demonstrating the Presence of a Third Variable 168
Table 4.5 Sample of the Results Obtained from Fuzzy Set Analysis 180
Table 4.6 A comparison between predicted and occurred events at two times t = 1 and 2 .. 186
Table 4.7. A comparison between suggested and implemented risk management practices 188
Table 4.8 Results of the Analysis of the three cases 191

List of Figures

Figure 1.1 Research Agenda 5
Figure 2.1. Research Method 24
Figure 2.2 The Three Core Processes of Software Projects 30
Figure 2.3 The Core Processes and Associated Outcomes of Software Projects 31
Figure 3.1 Conceptualizations of Software Project Risk in IS Literature 61
Figure 3.2 Procedure Used in Semantic Decompositional Analysis 67
Figure 3.3 Semantic Relations of Lexical Items 70
Figure 3.4 Action Verbs Lexical Items (Sentence Semantic Relations) 72
Figure 3.5 A Causal model of the Relationships between Residual Risk Components 79
Figure 3.6 Results of the Relationships between Residual Risk Components 88
Figure 4.1 Software Project Residual Risk (El-Masri and Rivard, Essay 2) 153
Figure 4.2 Research Methodology 155
Figure 4.3 Componentization of the Project Residual Risk Construct Dimensions 156
Figure 4.4 Interrelationships among the Dimensions of the Residual Risk Construct 160
Figure 4.5 Mangan and Kelly (2009) Case Configuration 178
Figure 4.6 Degree of Membership of Risk Components of Mangan and Kelly (2009) Case 179
Figure 4.7 A Subset of the Software Project Risk Management iThink Simulation Model . 181

Acknowledgements

Throughout my long journey writing this thesis, I was fortunate to have my director and mentor Professor Suzanne Rivard always on my side. Suzanne, you were my *sensei* for seven years, and for that, I owe you my deepest and most sincere gratitude.

I wish to thank my committee members Professor Henri Barki and Professor Liette Lapointe. Without your comments and guidance, I would not have completed this study. I also extend my gratitude to my colleagues, faculty members, thesis jury members, and all members of the information systems research group (GReSI) at HEC Montréal. You provided me with invaluable feedback and encouraged me to do the best I can.

My special thanks goes to Genevieve, my beloved wife, for her infinite support. You believed in me and were always there when I needed you most. I also want to thank my daughter Naomie and my son Ismaël for putting up with me all this time and for their unbounded love.

The strength that I needed to complete this thesis comes from my mother Aniseh. Mama, you keep on showing me how to overcome obstacles and achieve what I want. I will be forever indebted to you.

This thesis is dedicated to my deceased father.

Chapter I – Introduction

The failure to identify and manage the risk of software projects has often been described in information systems literature as a major detriment to their success (Wallace et al. 2004a). One approach that information systems researchers use to study this phenomenon is to identify risk factors that negatively correlate with project success (e.g., Barki et al. 1993; Schmidt et al. 2001) as well as risk management practices that mitigate their effects (e.g., McFarlan 1981; Tesch et al. 2007). In practice, risk is an important aspect of software projects that managers attempt to evaluate and control (Ropponen and Lyytinen 2000). Renowned institutes such as PMI that provide project management certifications instructing managers on risk assessment and control are continuously growing in membership (11% in 2011) (Partleton 2011). Those certifications teach managers how to apply project management techniques such as PERT and CPM that can help manage risk. Moreover, the use of project management software in order to plan and track project activities has been constantly growing (Liberatore and Pollack-Johnson 2003). Software and add-ons that are specific to risk management such RiskyProject Professional and Palisade @RISK have also been offered to support their risk management activities.

Nevertheless, in the past two decades the success rate of software projects has been relatively stagnant. Indeed, the CHAOS reports that frequently document the ratio of successful, challenged, and failed software projects present a grim reality. Between 1996 and 2011, success rates of software projects varied between 26% and 35% (Standish Group 2011). The 2011 CHAOS report shows that 51% of software projects are challenged and another 15% are a complete failure. These numbers are surprising considering the advances of software project management standards (e.g.; ITIL, PRINCE2, and Agile) and the rapid growth of certified project managers (a 100 fold increase in PMP certifications since the 1990).

More recently, exploratory research uncovered dynamic interactions between risk factors (e.g., Sicotte and Paré 2010; Warkentin et al. 2009). For instance, Warkentin et al. (2009) found that one recipe for failure is when management inadequately plans a project that is characterized as technically risk (complexity) and with requirement risk (instability, ambiguity, and mismatched user expectations). The authors emphasize the importance of assessing risk components simultaneously instead of independently. Interactions between risk factors were also examined over time. In a longitudinal multi-case study, Sicotte and Paré (2010) investigated the dynamic evolution of interdependence between risk factors. For example, the authors found interactions between human risk (e.g., resistance) and political risk (e.g., conflict of interest) that evolved over time leading the project to failure.

Nevertheless, risk management techniques that can take into consideration the interaction effects of risk on project success are lacking in research as well as in practice. The information systems literature does not provide a theoretical framework that can account for these interactions. Moreover, the risk management software that is available in the market to support project managers do not include features that can address this phenomenon. Taking the dynamic interactions into account can help us better understand risk and the risk management process. This constitutes the primary motivation for our research.

Research Objective

This dissertation puts forward the design principles for a software project risk management system that accounts for the interaction effects of risk components. Notwithstanding the fact that design science is concerned with the design and evaluation of information systems, the design of the artifacts depends on knowledge from natural science (Sicotte and Paré 2010; Warkentin et al. 2009). In this case, the requisite knowledge concerns the software project success and risk constructs and the interrelationships between their components. Therefore, the first two articles in this dissertation refines the two constructs by taking into consideration the diverse conceptualizations provided in the IS literature.

Research Question

The research questions that design science aims to answer are different than those in natural science. Design science is the “science of the artificial” (Gregor and Jones 2007). Contrary to natural science research that seeks the truth about certain phenomena, the goal of design science research is utility (Simon 1996). The questions that design science seeks to answer are whether the artifact design works and whether it is an improvement from prior methods or systems (Hevner et al. 2004; Pries-Heje and Baskerville 2008). Consequently, this dissertation answers the following research questions:

1) What are the necessary features of a software project risk management system that can account for the interrelations between risk components?
2) Given an instantiation of an information system that account for such interrelations:
    a. Is its predictive capability supported by empirical evidence?
    b. Is its usefulness supported by empirical evidence?

Research Methodology

Several methods are used in order to achieve the research objectives (see figure 1). The objective of the first article is to specify a comprehensive software project success construct that is consistent with its various conceptualizations found in the information systems literature. To this end, a process perspective is adopted to reorganize 23 years of relevant information systems literature according to three core software project processes and their associated outcomes. The success construct is then extended to incorporate the stakeholder perspective and the existing IS literature is reorganized accordingly.

The objective of the second article is to develop and validate a comprehensive software project residual risk construct that integrates the existing theoretical perspectives of risk in the information systems literature. More specifically, we seek to combine the core risk components of existing risk conceptualizations and recognize how they interrelate. To accomplish this objective, we analyzed the semantic structure of existing definitions of software project risk using a semantic decomposition technique in order to reveal its various sub-concepts and how they relate to one another. Then, we reconstructed the text into a single comprehensive definition. We used this definition to conceptualize a multi-dimensional software project residual risk construct and mathematically express its measurement. To validate that the proposed construct is adequately designed and the relationships between its dimensions are plausible, we evaluated a causal model that reflects its structure. More specifically, the dimensions of residual risk were defined as constructs in a model and the relationships between them were defined as paths. Expert judges helped us specify the construct's measurement model from existing risk items in the information systems literature. The measurement model was used as a coding scheme in a case survey in order to provide the necessary data to test hypotheses on the relationships between the model's variables. Partial least square (PLS) was used to analyze the data.

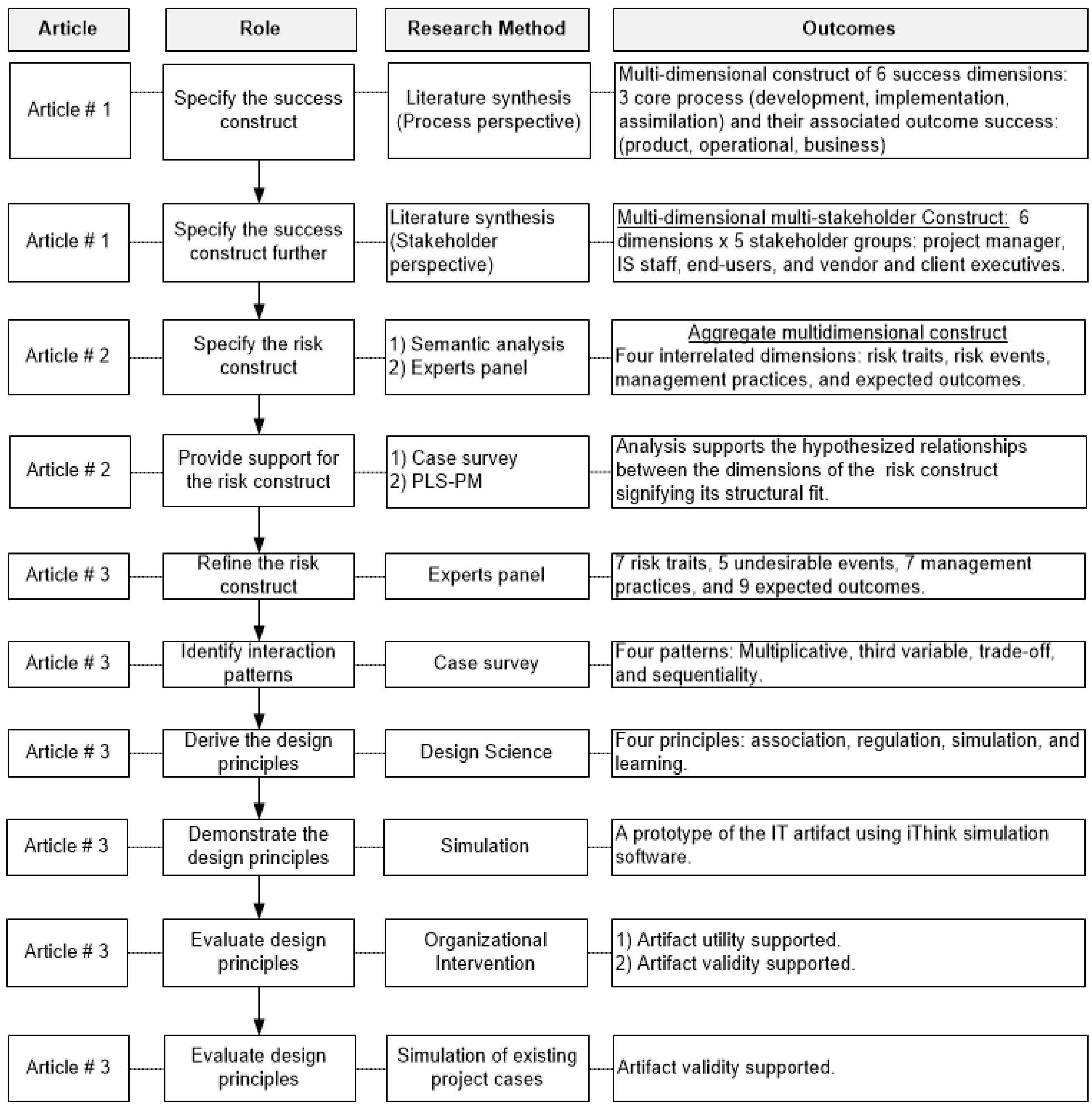


Figure 1.1 Research Agenda

The third paper develops, demonstrates, and evaluates the design principles of an IT artifact that accounts for the risk interplay phenomenon. To discover the design principles, the residual risk construct developed in the second article was refined in two stages. Initially, a panel of judges reclassified the risk items identified from the

IS literature under different components of risk traits and undesirable events that we provided. The working residual risk construct was also refined by identifying patterns of interrelationships between its components. To this end, a case survey was conducted. From those patterns, a set of design principles was derived. The design principles were reflected in an IT artifact that was developed using simulation software. To evaluate the IT artifact, and thereby the design principles, an intervention in the risk management of a software project was conducted. Cases from the IS literature were also used to simulate risk scenarios and provide further support for the IT artifact's validity.

Motivation and Contribution

Investments in information technology continue to occupy a significant sum of organizational spending. On average, organizations allocate 4% of their revenue for IT of which 22% is invested in software products (March and Smith 1995). Increasingly, organizations are applying project-based management approaches in order to manage those expenditures (Gartner 2010). Management by projects has often been associated with improved control and rapid market response (Reich 2007). Yet, most organizations still find it difficult to run software projects successfully (Reich 2007). The 2011 CHAOS report affirms that the majority of software projects are still unsuccessful (Faraj and Sambamurthy 2006).

In light of recent findings pertaining to risk factor interactions (Standish 2011), the community of practice can benefit from a new information system that accounts for this complexity of software project risk. If neglected, the rippling effect of interplay between risk components can result in the underestimation of a project's risk exposure and render the project uncontrollable. Since no available project risk management software addresses the interrelations of risk factors, the design science paradigm is judged to be an appropriate lens since it can provide a practical solution to project managers.

Academically, the dissertation answers the call to reengage the IT artifact in IS research. According to Benbasat and Zmud (Sicotte and Paré 2010; Warkentin et al. 2009), IS research should focus on the development of practices to plan, construct, and implement IT artifacts. The design theorizing approach to the development of the IT artifact advances knowledge of software project risk management while providing a practical solution to managerial problems. Moreover, the specification of software project success that we arrive at in the first paper integrates the stakeholder's perspective while taking into account the main processes and outcomes of software projects. Taking a process perspective, the model distinguishes between the three core project processes and their ensuing outcomes. This helps clarify the ambiguities of existing conceptualizations in the literature that specify project success according to the process and outcome dimensions. Also, the residual risk construct specified in the second paper classifies the large number of risk factors previously identified in the literature under four dimensions. This classification eliminates redundant factors that were once aggregated under a single risk factor list, thereby improving the accuracy of the software project risk exposure scale.

Paper 1: Specifying Software Project Success

The clarification of the software project success construct receives a great deal of attention in the IS field. Predominantly, the construct has been conceptualized in three different ways: (1) with respect to the cost, time, and quality/scope (2) in terms of process and outcome success dimensions and, (3) according to a list of criteria that researchers consider sufficient to capture project success. Overall, the existing conceptualizations have been criticized (e.g., Agarwal and Rathod 2006; Linberg 1999; Ojiako et al. 2008). A review of the literature shows that there is a need for a success construct that theoretically organizes the different facets of software project success as well as considering the perspectives of key project stakeholder groups. Although the conceptualization of success in terms of the process and outcome dimensions is well recognized (e.g., Aladwani 2002a; Nidumolu 1995; Tiwana 2012), much remains unclear about how these dimensions are defined and specified. Moreover, the stakeholder perspective has not been integrated to existing conceptualizations. The literature provides different project stakeholder perceptions when asked to appraise the success of their software projects (Bartis and Mitev 2008).

Research Objective

In this paper, we take a process view and derive a comprehensive software project success construct from the different conceptualizations in the IS literature while accounting for the stakeholder perspective.

Research Method

A literature review is conducted to conceptualize and specify the software project success construct. We reviewed 23-years (1990-2012) of the relevant literature published in MIS Quarterly, Information Systems Research, Journal of Management Information Systems, and the International Journal of Project Management. We also searched the ABI/Inform and Business Source Premier databases for additional literature. A total of 144 articles are identified.

Conceptual Model

Software project success is conceptualized as the combination of the successes of three core processes and associated outcomes as perceived by different project stakeholder groups. The processes are the development, implementation [1], and assimilation processes, whereas the outcomes relate to the product, organization, and business. Various stakeholder groups perceive the success of each of these six dimensions differently. Accordingly, the construct also delineates what success signifies for each stakeholder group; the project manager, the IS staff, the end-users,

[1] We adopted Alioni et al.'s (2007) definition of software implementation throughout the thesis. Accordingly, this process is restricted to "activities from software deployment or installation to parameterization, integration, testing, and stabilization".

the client executives, and the vendor executives. This construct refinement was derived from research that examines the different perceptions of key project stakeholder groups.

Contribution

This paper conceptualizes software project success as a multi-dimensional multi-stakeholder construct and specifies it in this respect. The resulting model extends the extant literature in two different ways. First, it presents a disciplined model that accounts for the diverse facets of software project success yet remains consistent with the conceptualization of success in terms of process and outcome dimensions. Second, it takes a stakeholder's perspective to present a multi-stakeholder construct that helps classify the software project success criteria according to the different stakeholder groups.

Paper 2: Specifying Software Project Residual Risk

The existing IS research conceptualizes software project risk in three different ways: (1) as a set of risk factors – also referred to as states, aspects, properties, events, or contingencies, (2) as a function of the probability of occurrence of undesirable events (or outcomes) and the consequences of the occurrence of these events (or outcomes), and (3) as a variance of project outcomes. Notwithstanding the merit of the various conceptualizations in enriching the knowledge about software project risk, this diversity reflects in disparate specifications of the risk construct. Moreover, while risk factors represent an essential constituent of the vast majority of the software project risk literature, their nature varies across the different operationalizations. Specifically, risk factors are often defined as conditions or attributes that represent a threat to software project success. Still, undesirable events are used as risk factors even if they are explicitly defined as project attributes (Bartis and Mitev 2008).

Research Objective

The objective of this article is to develop and evaluate a comprehensive software project residual risk construct and define its primary dimensions and relationships by integrating the existing theoretical perspectives of risk in the information systems literature.

Research Method

Construct development was achieved by conducting a semantic decomposition analysis on existing risk definitions and other descriptive texts found in the IS literature. The review covered 23 years of software project risk literature (1990-2012). The eight leading information systems journals, project management journals, and management-related journals published by the Institute of Electrical and Electronics Engineers were included. To provide support for the construct's structure and the relationships between its dimensions, we developed and evaluated a risk model that mirrors the construct's internal structure and the relationships between its dimensions. A case survey of 82 software project cases was conducted to gather data and test hypotheses on the relationships between the model's constructs. The construct's dimensions were used as the case survey's coding scheme categories. A panel of experts helped specify the categories by reclassifying the existing project risk items accordingly. The dataset that resulted from the case survey was statistically analyzed with partial least square (PLS).

Conceptual Model

The semantic decomposition analysis resulted in the identification of four main dimensions of software project residual risk: risk traits, undesirable events, expected outcomes, and risk management mechanisms. The analysis also indicated how the four dimensions relate to one another. Accordingly, software project residual risk was defined as the conditional probability that project risk traits will cause undesirable events resulting in a negatively valued deviation from the expected project outcomes given that all accessible risk management practices are considered.

Contribution

From a research perspective, the proposed risk construct is comprehensive yet parsimonious. Hence, it can be adopted in future studies regardless of their theoretical perspective. It can also organize the large number of risk factors found in the IS literature under its four dimensions. This helps eliminate redundant factors that were formerly put under a single list. The construct also answers research calls to integrate risk management practices as a central dimension. For practice, the risk construct can be used as a schema managers use to identify project risk traits.

Paper 3: Towards a Design Theory for Software Project Risk Management Systems

The IS research community reveals an intricate and dynamic representation of software project risk. According to the empirical studies of Sicotte and Paré (e.g., Ropponen and Lyytinen 2000) and Warkentin et al. (2010), software project risk factors are closely intertwined. Over time, their interaction effects progressively inflate a project's exposure to risk which can result in adverse outcomes. While prior research has called for the investigation of risk factor interrelations (e.g., Alter and Sherer 2004; Schmidt et al. 2001), only recently did the exploratory studies of Sicotte and Paré (2009) and Warkentin et al. (2010) provide support for this phenomenon. Yet, little is known about the nature of these interrelationships. An investigation of the project risk management software available for practitioners reveals that existing tools provide only basic functionalities and are not tailored to address such particularities of software project risk.

Research Objective

This article defines, demonstrates, and evaluates the design principles of a software project risk management system that accounts for the interactions among risk components.

Research Method

A Design science research approach (DSR) is adopted to achieve the above research objectives. DSR complements and extends IS behavioral-science research by creating and evaluating IT artifacts that reflects existing theoretical foundations. The methodology process model articulated by Peffers et al. (2007) was followed which primarily consists of three stages: development, demonstration, and evaluation. To develop the design principles, the refinement of the residual risk construct from the second article was necessary to allow for the discovery of the interrelationships between its components. To this end, another panel of judges helped reclassify existing risk items under components of the risk traits and undesirable events dimensions that were provided to them. The patterns in which components of risk interrelate were identified by conducting a case survey. Both the identified patterns and the refined risk construct were interpreted to derive a set of design principles of the IT artifact. To demonstrate the design principles, an instantiation of an IT artifact (a prototype) that embeds those principles in its form and function was designed. System dynamics-based simulation modeling software (iThink) was used to mathematically model the interlocking interrelationship between risk components. The dataset generated from the case survey was analyzed using the fuzzy-set approach to determine the strength of relationships between the risk components. To evaluate the IT artifact's utility and validity, an intervention in the risk management of a software implementation project was conducted in an organizational setting. We also ran simulation scenarios based on the characteristics of three software project cases from the IS literature and compared the accuracy of the IT artifact's predictions with the cases' realities.

Contribution

Software project risk management is an iterative process of risk assessment and response Peffers et al. (2007). During this process, the evaluation of project residual risk exposure is determined using an intricate and active function of interrelated variables of different natures. Those variables belong to four categories: risk traits, undesirable events, managerial practices, and the expected outcomes. Hence, reducing risk to an inert variable to facilitate its quantification, as is the case in variance models, reduces its applicability in practice. When construct specification is complex, design research as an applied science is a good alternative to natural science. Using the design science paradigm, the specification of the software project residual risk construct becomes a creative design process that is informed by knowledge from natural science and guided by academic methods of design theory development.

The relevance of design science research is primarily related to its direct impingement on practice (Boehm 1991). As the intervention results suggest, the artifact provides project managers with an improved practical solution to assess and control the risk of software projects. When grounded in theory, design science research advances academic knowledge while keeping the scientific integrity of its outcomes. Making the IT artifact the focus of scientific research advocates that the risk management phenomenon can be examined using scientific theories and that the research outcomes can improve practice.

References

Agarwal, N., and Rathod, U. 2006. "Defining 'success' for software projects: An exploratory revelation," *International Journal of Project Management* (24:4), pp 358-370.

Alter, S., and Sherer, S., A 2004. "A General, But Readily Adaptable Model of Information System Risk," *Communications of the Association for Information Systems* (14:1), pp 1-28.

Barki, H., Rivard, S., and Talbot, J. 1993. "Toward an Assessment of Software Development Risk " *Journal of Management Information Systems* (10:2), pp 203-225.

Bartis, E., and Mitev, N. 2008. "A multiple narrative approach to information systems failure: a successful system that failed," *European Journal of Information Systems* (17:2), pp 112-124.

Benbasat, I., and Zmud, W. R. 2003. "The Identity Crisis within the IS Discipline: Defining and Communicating the Discipline's Core Properties," *MIS Quarterly* (27:2), pp 183-194.

Boehm, B. 1991. "Software Risk Management: Principles and Practices," *IEEE Software* (8:1), pp 32–41.

Faraj, S., and Sambamurthy, V. 2006. "Leadership of Information Systems Development Projects," *IEEE Transactions on Engineering Management* (53:2), pp 238-249.

Gartner 2010. "IT Metrics: IT Spending and Staffing Report."

Gregor, S., and Jones, D. 2007. "The Anatomy of a Design Theory.," *Journal of the Association for Information Systems* (8:5), pp 312–335.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design science in Information Systems Research.," *MIS Quarterly* (28:1), pp 75–105.

Liberatore, M. J., and Pollack-Johnson, B. 2003. "Factors influencing the usage and selection of project management software," *Engineering Management, IEEE Transactions on* (50:2), pp 164-174.

Linberg, R. K. 1999. "Software developer perceptions about software project failure: A case study," *The Journal of Systems and Software* (49:2), pp 177-192.

March, S. T., and Smith, G. F. 1995. "Design and Natural Science Research on Information Technology," *Decision Support Systems* (15:4), pp 251-266.

McFarlan, F. W. 1981. "Portfolio Approach to Information Systems," *Harvard Business Review* (59:5), p 142.

Ojiako, U., Johansen, E., and Greenwood, D. 2008. "A qualitative re-construction of project measurement criteria," *Industrial Management + Data Systems* (108:3), pp 405-417.

Partleton, B. 2011. "PMI 2011 Annual Report: Delivering Value," in *PMI*, PMI.

Peffers, K., Tuunanen, T., Rothenberger, M., and Chatterjee, S. 2007. "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems* (24:3), pp 45-77.

Pries-Heje, J., and Baskerville, R. 2008. "The Design Theory Nexus," *MIS Quarterly* (32:4), pp 731-755.

Reich, B. H. 2007. "Managing Knowledge and Learning in IT Projects: A Conceptual Framework and Guidelines For Practice " *Project Management Journal* (38:2), pp 5-17.

Ropponen, J., and Lyytinen, K. 2000. "Components of Software Development Risk: How to Address Them? A Project Manager Survey," *IEEE Transactions on Software Engineering* (26:2), pp 98-111.

Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. 2001. "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems* (17:4), pp 5-36.

Sicotte, C., and Paré, G. 2010. "Success in Health Information Exchange Projects: Solving the Implementation Puzzle," *Social Science & Medicine* (70:1), pp 1159-1165.

Simon, H. A. 1996. *The Science of the Artificial*, (Third ed.) MIT Press: Cambridge, MA.

Standish, G. 2011. "Standish Group Chaos Report, CHAOS Summary 2011," Standish Group.

Tesch, D., Kloppenborg, J. T., and Frolick, N. M. 2007. " IT Project Risk Factors: The Project Management Professionals Perspective," *The Journal of Computer Information Systems* (47:4), p 61.

Wallace, L., Keil, M., and Rai, A. 2004. "How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model*," *Decision Sciences* (35:2), pp 289-289-321.

Warkentin, M., Moore, R. S., Bekkering, E., and Johnston, A. C. 2009. "Analysis of Systems Development Project Risks: An Integrative Framework," *Database for Advances in Information Systems* (40:2), pp 8-27.

Chapter II – Article 1

SPECIFYING SOFTWARE PROJECT SUCCESS

Mazen El-Masri
HEC Montréal
3000 Chemin-de-la-Côte-Ste-Catherine
Montréal, CANADA, H3T 2A7
Telephone number: 514 340-6000 ext. 2553
Fax Number: 514 340-6132
Mazen.el-masri@hec.ca

Suzanne Rivard
HEC Montréal
3000 Chemin-de-la-Côte-Ste-Catherine
Montréal, CANADA, H3T 2A7
Telephone number: 514 340-6493
Fax Number: 514 340-6132
Suzanne.rivard@hec.ca

Abstract

This paper combines the process and stakeholder perspectives to develop a model of the software project success construct. We analyze 23 years of research pertaining to software project success and reorganize it. First, we propose three core processes within the software project lifecycle and reclassify extant success criteria accordingly. Consequently, the literature on stakeholder perceptions is examined in order to identify what success means for each group in regards to the three project processes and their outcomes. The working model is then extended to incorporate the stakeholder dimension and refine our classification of the extant success criteria. Lastly, we discuss the implications to research and practice of conceptualizing software project success as the combination of the successes of a *series* of processes and associated outcomes as perceived by the different project stakeholder groups.

Keywords

Software project, software project success, process model, stakeholder's perspective

Introduction

Software project success is an important concept for research and practice. In research, studies examine how it relates to other concepts including planning (Aladwani 2002b), end-user involvement (Wang et al. 2005), coordination (Andres and Zmud 2002), risk (Na et al. 2007), and risk management (de Bakker et al. 2012). In practice, software project success is a fundamental constituent of today's economy. Organizations are increasingly managing their information systems initiatives and operations using project-based management approaches (Pant and Baroudi 2008; Reich 2007). Managers apply best practices like PMBOK, ITIL, and Agile (e.g., PMI 2004; Van Bon et al. 2007) to increase their projects' chances of success. Yet, they still see over 63% of these projects as being challenged or having failed altogether (Standish Group, 2011).

Knowing what success signifies is a precondition of successful projects. In the IS literature, there exist a number of conceptualizations of software project success. One way success is viewed is according to cost, duration, and product quality (e.g., Abdel-Hamid et al. 1999; Mahaney and Lederer 2011; Warkentin et al. 2009). Some research broadens the notion of success by complementing these three criteria with others like system use (Atsu et al. 2010), organizational benefits (Nicholas and Hidding 2010), user satisfaction (Hahn et al. 2009), and team morale (Liu et al. 2011b). Other pertinent research provides a theoretical success framework and conceptualizes it using two-dimensions – process and outcomes (e.g., Aladwani 2002a; Nidumolu 1995; Tiwana 2012). According to this view, software projects are successful if they are managed efficiently and produce effective results (Aladwani 2002a).

The software project success construct also receives the attention of researchers who try to clarify or define it. Some studies argue that limiting success to only three criteria – cost, duration, and quality – does not provide a true picture of how projects perform (Agarwal and Rathod 2006). While augmenting the cost-duration-quality view of success to include additional criteria extends the success construct's coverage (e.g., Liu et al. 2011b; Tiwana 2012; Xu et al. 2010), it seems insufficient to capture the software project's many success facets. Projects can complete on-time, on-budget, produce high quality and usable software, and bring value to organizations, yet their stakeholders (manager, team, users, sponsors, and top management) can still consider them failures (see: Nelson 2005). Likewise, projects can fail to achieve the three criteria and be considered a success (ibid). Even canceled projects are deemed successful if they provide value like learning for subsequent projects (ibid).

Although viewing success as a multi-dimensional construct of process and outcome(s), provides a useful theoretical framework to model its multiple facets, the extant conceptualizations are diverse. Some view process success in terms of management's ability to control the process, learn, and interact with end-users (e.g., Liberatore and Luo 2010; Nidumolu 1995). Others view it in terms of management's ability to respect the three criteria (e.g., Nelson 2005; Nicholas and Hidding 2010). Likewise, software project outcome success has been conceptualized differently. For some, outcome success is viewed from a technical perspective and conceptualized in terms of software quality and technological capability (e.g., Wallace et al. 2004b). Others view outcome success from a financial perspective and conceptualize it in terms of improved organizational financial and strategic performance resulting from the use of the resulting product (e.g., Kearns 2007; Nicholas and Hidding 2010). Moreover, some criteria like software quality have been viewed as part of both process success (e.g., Nelson 2005) and outcome success (e.g., Cooke-Davis 2002; Wixom and Watson 2001).

Information systems research also emphasizes the importance of considering the stakeholder perspective when conceptualizing software project success (e.g., Agarwal and Rathod 2006; Ambler 2007; Linberg 1999; Ojiako et al. 2008).

According to Freeman (2010), management can create better value when the interests of different stakeholders are considered. An incomplete knowledge of those interests may result in misunderstanding the stakeholders' perception of success, and lead to unexpected overall outcomes (Sherif et al. 2006). Differences in stakeholders' perceptions of software project success have been emphasized in IS research. For example, Bartis and Mitev (2008) show two different stakeholder groups who simultaneously recognize a project both as a success and a failure. Other studies describe differences in stakeholder perceptions of project success (e.g., Ambler 2007; Linberg 1999; Ojiako et al. 2008). Executive managers define success in terms of realizing strategic objectives while project managers emphasize respecting the cost, time, and quality criteria (Ojiako et al. 2008; Shenhar et al. 2002). Conversely, IS staff tend to view success according to software quality, project management quality, personal achievement, professional development, and learning (Linberg 1999), while end-users consider a project successful if the software satisfies their operational needs (Nelson 2005).

This paper proposes a conceptualization of software project success as a combination of the successes of a series of processes and their associated outcomes as perceived by different project stakeholder groups. By distinguishing various project phases and stakeholders, a clear and comprehensive success construct can be developed that also answers research calls to incorporate stakeholder perspectives.

Software projects go through various phases from the time they are initiated until the software reaches normal use. The objectives of each phase vary depending on whether the outcome of interest is the software itself, its use, or its benefits. To better understand what success signifies in each phase, both from the process and outcome perspectives, it is necessary to distinguish these phases and their core processes. The conceptualization of success in terms of a combination of interrelated processes and outcomes can provide a grounded structure to organize the large number of success criteria found in the literature. Likewise, it is important to consider the relevance of each phase to the different stakeholder groups. Stakeholders' expectations could also vary by stage. For instance, in the development stage, learning is one of the important success criteria for project team members (Procaccino et al. 2005). When the software reaches the delivery stage, team members evaluate success according to the satisfaction of end-users with the quality of the product (Wateridge 1998). Adopting the stakeholder perspective to distinguish stakeholders' views of success at each stage can explain why failed projects are often perceived to be successful and vice versa. It could help managers recognize what each project stakeholder group expects and at which stage of the software project.

The rest of the paper is organized as follows. First, we explain the approach adopted to identify the relevant literature. Subsequently, the different conceptualizations of software project success found in the literature are delineated. The literature is then reorganized according to three core processes and associated outcomes of the software project lifecycle. Then, we extend our model to incorporate the stakeholder perspective and reorganize the literature accordingly. We finish by discussing our proposed conceptualization of the software project success construct and its implications for research and practice.

Identifying The Relevant Literature

The review of the literature covered 23 years (1990-2012) of IS research. We examined the abstracts of articles published in three IS journals – MIS Quarterly, Information Systems Research, and the Journal of Management Information Systems – and retained those that discussed the notion of information systems project success. We made no distinction between software projects and IS projects. The literature searched was not restricted to IS journals. The International Journal of Project Management was also searched for articles that discussed IS project success. Then, to insure a wider coverage of the IS literature, we used keywords to search the ABI/Inform and Business Source Premier databases for articles from other journals.

Articles that had “information systems”, “software”, or “information technology” as a subject and “project success”, “project management”, “project performance”, “project failure”, and “implementation success” in their title or abstract were checked. Finally, we identified eight key articles. The chosen articles (see Appendix A) addressed the conceptualization of software project success and were deemed essential to our study. Therefore, we searched the Web of Science database to find studies that referenced them. The relevant studies we found were added to our set of articles. The search resulted in 144 usable articles (table 2.1) published in North American and international journals (105 and 39 respectively). A large number of articles published in North American journals collected data in countries like Australia, UK, Chile, Spain, and Taiwan.

Table 2.1 Number of Articles Identified by Journal

Name of Journal	**Number of Articles**
International Journal of Project Management	24
MIS Quarterly	19
Journal of Management Information Systems	18
Information Systems Research	15
Project Management Journal	11
Information & Management	9
The Journal of Systems and Software	8
The Journal of Computer Information Systems	5
Information and Software Technology	4
Communications of the ACM	4
European Journal of Information Systems	4
IEEE Transactions on Engineering Man. (and Software Engineering)	4
Information Systems Management	3
Other journals	16
Total	**144**

Research Method

The research approach we adopted to organize the identified literature and provide a conceptualization of software project success comprised three stages (see figure 2.1). In the first stage, the articles were analyzed and grouped according to their authors' conceptualizations of software project success. In the second stage, a process approach was applied to organize the literature. We reverted to the IS literature to identify the core processes and associated outcomes of the software project lifecycle model. This working model was then used to classify the success criteria we had identified in the literature under its different processes and outcomes. The third stage extended the working model by integrating five key stakeholder groups as a second dimension. This working model was then used to reclassify the success criteria under the five stakeholder groups.

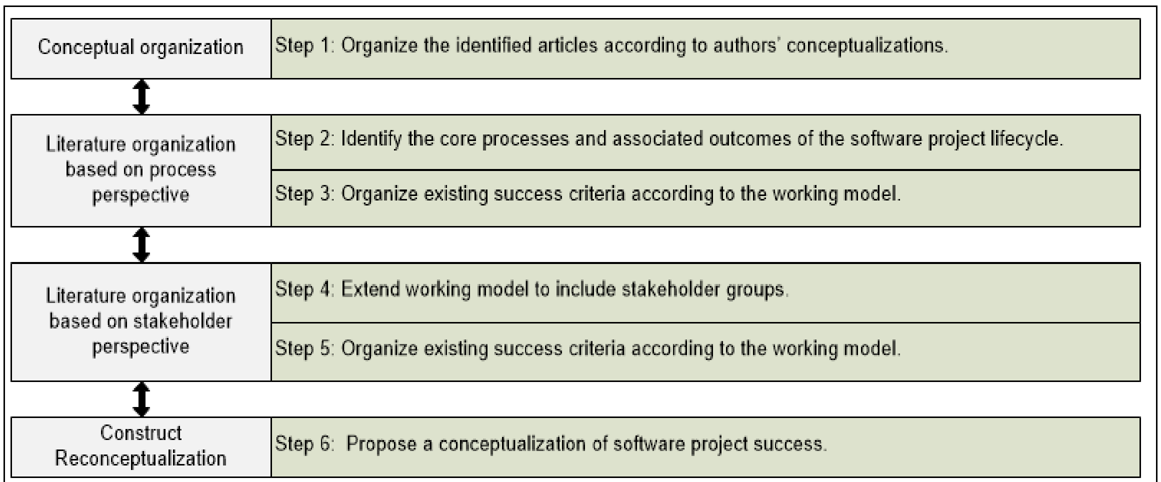


Figure 2.1. Research Method

Conceptualizations of Software Project Success

The identified articles were analyzed and organized depending on their conceptualization of software project success (see table 2.2). Four groups were identified. The first group conceptualized success in terms of time, cost, and quality. Some of these articles considered completing the project on-time, on-budget, and according to scope (or functionality) to be the three criteria of software project success (e.g., Banker et al. 2006; Lee and Xia 2010). However, the majority of articles that belong to this group chose achieving the intended software quality as the third criteria instead of scope (e.g., Mahaney and Lederer 2011; Yetton et al. 2000).

Table 2.2 Organization of the Identified Literature

Software Project Success	**No. of Articles**	**Percentage**
On-time, on-budget, and to quality	25	17%
A list of criteria	59	41%
Process and outcome dimensions	51	36%
Stakeholder perspectives	9	6%
Total	**144**	**100%**

The second group extended the cost, time, and quality conceptualization to include additional success criteria. This second conceptualization is the most common in the information systems literature (see table 2.2). Some criteria authors included relate to work performance during project execution such as the efficiency of team operations (Aladwani 2002a), work morale (Chang et al. 2010), amount of work produced (Robey et al. 1993), and effectiveness of interactions with non-team members (Henderson and Lee 1992). Satisfaction-related criteria such as the satisfaction of end-users (Xia and Lee 2004), customers (Thite 2000), and team members (Alberto Espinosa et al. 2011) have also been appended to the traditional three success criteria. Other articles included criteria related to the software product itself such as usability (Basten et al. 2011) and flexibility (Liu et al. 2011a). Some of the research belonging to this group also extended the list to include operational criteria such as system use (Hahn et al. 2012) and minimum user resistance (Naveed 1996). The list of criteria has been extended even further to include financial and strategic performance criteria such as business value (Sauer et al. 2007) and market responsiveness (Dvir et al. 2003) and revenue (Thong et al. 1996).

Table 2.3 The Nature of Success Criteria of the Second Group of Articles

Nature of Success Criteria	**Examples of Criteria Used**	**References**
Project performance	Efficiency of team operations, high work morale, amount of work produced, effectiveness of interactions with non-team members.	Aladwani (2002a), Chang et al. (2010), Chen et al. (2011), Henderson and Lee (1992), Hsu et al. (2012), Robey et al. (1993), Xu et al. (2010).
Product performance	Usability, quality, ease of use, response time, flexibility.	Atsu et al. (2010), Basten et al. (2011), de Bakker et al. (2012), Liu et al. (2011a).
Operational performance	System use, minimum user resistance, superior performance.	Hahn et al. (2012), Naveed (1996).
Business performance	Business value, market responsiveness, revenue.	Dvir et al. (2003), Sauer et al. (2007), Thong et al. (1996).
Satisfaction	End-users satisfaction, customer satisfaction, team satisfaction.	Xia and Lee (2004), Thite (2000), Alberto Espinosa et al. (2011).

The third group conceptualized software project success as comprising a process(es) and an outcome(s) dimension (e.g., Deephouse et al. 1996; Tiwana 2012). Process success defines how well a project has been carried out while outcome success describes how well the delivered software performed (Nidumolu 1995; Procaccino et al. 2005). Some authors see process success in terms of management effectiveness and define it according to budgetary and schedule criteria (see table 2.4). Others view process success in terms of management efficiency and conceptualized it based on predictability and control over resources and flexibility in the face of labor and market fluctuations (Nidumolu and Subramani 2003). Process success is also viewed in terms of efficiency and producing the intended work with minimum rework (Ravichandran and Arun 2000). Some research that investigates software implementation projects (e.g., ERP) viewed a successful process to include successful data conversion and integration with third party software (Hakkinen and Hilmola 2008). Minimizing end-user resistance and disturbance to ongoing operations were also considered as elements of an efficient management process (Markus and Tanis 2000). Studies that investigated implementation projects also assessed process success according to end-users' degree of use (Liang et al. 2007).

Table 2.4 Conceptualizations of Process Success

Conceptualization	**Examples of Variables Used**	**References**
Project management effectiveness	On-time and on-budget.	Barki et al. (2001), Thomas and Fernández (2008), Nelson (2005).
Project management efficiency	Controllability and predictability over resources	Nidumolu and Subramani (2003), Gopal and Gosain (2010)
	Minimizing rework and defects	Ravichandran and Arun (2000).
	Quality of interactions, efficiency of operations.	Aladwani (2002a).
Implementation effectiveness	Data conversion, 3rd party integration, system use.	Hakkinen and Hilmola (2008), Liang et al. (2007), Sharma and Yetton (2007).
Implementation efficiency	Minimizing user resistance and disturbance of ongoing operations	Markus and Tanis (2000).

Conceptualizations of outcome success vary depending on the authors' outcome of interest. When the project outcome of interest is the developed software, outcome success was viewed according to software quality and functionality (see table 2.5). However, when the outcome of interest was the software in operation, outcome success was viewed in terms of use, user satisfaction, productivity, and operational goals. Realizing business objectives such as ROI, revenue and strategic benefits have also been measures of outcome success. A number of articles split outcome success into multiple dimensions. For instance, Barki and Hartwick (2001) viewed outcome success according to three dimensions: software quality, organizational performance, and individual performance. Shenhar et al. (2002) saw the benefit to the customer, benefit to the organization and future potential as the three dimensions of outcome success.

Table 2.5 Conceptualizations of Outcome Success

Outcome of Interest	**Conceptualization**	**References**
Software developed	Quality and functionality.	Barki et al. (2001), Saarinen (1990)
Software in operations	Use, user satisfaction, operational performance	Nelson (2005), Wixom and Watson (2001)
Software in realization	Financial and strategic benefits	Dvir et al. (2003), Deutsch (1991)

Finally, the fourth group of articles aimed at clarifying the success construct using the stakeholder perspective. Project stakeholders are persons or groups that are actively involved in a project or they are those whose interests could be affected by the success or failure of the project (Ojiako et al. 2008). Generally, these articles highlighted differences in stakeholder perceptions of success and proposed different success criteria pertaining to each stakeholder group. They provide useful examples in which different stakeholder groups see the same projects as both successes and failures (e.g., Ojiako et al. 2008). Moreover, this group of research emphasizes the need to elaborate the stakeholder satisfaction variables often used as surrogates to measure success (e.g., Baccarini, 1999) to include targeted measures like learning, personal benefits, and professional development (Pereira et al. 2008).

A Process Perspective of Software Project Success

Our analysis of the different success conceptualizations, dimensions, and criteria compelled us to reexamine the success concept from a process perspective. It was evident that the studies that extend the traditional cost-budget-quality view of success to include other criteria attempted to capture success at various stages without recognizing how these stages are interconnected. Therefore, we identified three core processes and associated outcomes that software projects go through and reorganized the extant criteria accordingly.

Identifying the Core Processes of the Software Project Lifecycle

The success criteria that information systems studies adopt to specify the software project success construct depend on the type of software project examined. Predominantly, information systems researchers examine two types of software projects: software development projects and implementation projects of packaged software (e.g., ERP systems). The development process pertains to the creation, modification, and renewal of software products (Cockburn 2002). On the other hand, research that investigates implementation projects of packaged software generally considers a staged process that starts with software package selection and implementation and ends with software assimilation use (Liang et al. 2007; Markus and Tanis 2000). The implementation process corresponds to the project phase of deployment, parameterization, integration, testing, and stabilization of software products (Alioni et al. 2007). The assimilation process signifies the process of early software use, diffusion, and routinization of the installed software across the organization (Markus et al. 2000; Purvis et al. 2001). A phase of normal operations follows the assimilation process in which a time lag that can be in years is necessary to evaluate the financial and strategic objectives of the software project (Markus et al. 2000). During the normalization phase, the software ceases from being part of the software project and becomes a constituent of the client's daily operations.

Figure 2.2 The Three Core Processes of Software Projects

Research that investigates software development projects often includes post development criteria like organizational benefits to specify the project success construct (e.g., Espinosa et al. 2006; Jiang et al. 2001), but ignores post development processes. Organizational benefits are influenced by the assimilation process (Armstrong and Sambamurthy 1999). Yet, research that investigates software implementation projects ignores the development process even though over 67% ERP implementations require development efforts like custom modules and add-ons to the packaged system (Law and Ngai 2007). Therefore, the evaluation of software project success must take into account the three core processes projects can typically go through: development, implementation, and assimilation.

According to Mason (1978), there exist different levels of outputs of information systems which lead to different approaches researchers have taken to examine IS influence. Adapting communication theory, Mason explains that information "flows through a series of stages from its production through its use or consumption to its influence on individual and/or organizations". Each core process of the software project produces a particular outcome (figure 2.3). The development process produces a software product developed according to client specifications

(Cockburn 2002). As for the implementation process, it is intended to deploy, configure, and parameterize the new software according to the client's specifications and to integrate it with existing software (Alioni et al. 2007). At the end of the implementation process, the software is ready to be used. The assimilation process is intended to diffuse the information system across the organization and routinize its use (Purvis et al., 2001). At the end of this stage, the organizational operations are transferred to the new system and routinized across the different organizational departments. The degree to which the information system has been diffused in the assimilation process has an impact on the operational success of the client organization (Liang et al., 2007).

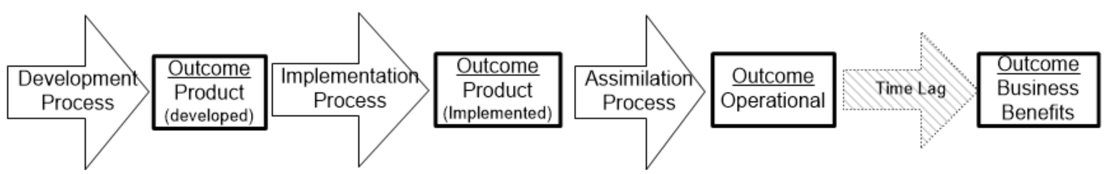


Figure 2.3 The Core Processes and Associated Outcomes of Software Projects

Only after the product is operational for a period of time can the client organization assess the impact of the software product on its financial and strategic objectives. In the implementation literature, the diffusion and routinization of the information system are necessary antecedents to business success (Ravishankar et al. 2011). Operational success, however, might or might not be translated to business success. They are, as Markus et al. (2000) describe them, early operational measures of success that loosely relate to the business and strategic objectives of the organization.

Organizing the Existing Success Criteria

The software project lifecycle model as comprised of three core processes and their associated outcomes can be used to conceptualize project success. Both the development and implementation processes are prerequisites of a successful product. The operational and business success of the software product comes after the assimilation process. In order to provide support for the conceptualization of the software project success construct, the success criteria found in the literature were classified under its six dimensions (see table 2.6).

The success of the software development process may perhaps be one of the aspects of software project that is examined the most in information systems research (e.g., Procaccino et al. 2005; Robey et al. 1993). The main success criteria pertaining to the software development process success dimension concern the team's ability to complete the planned development activities within the budgetary and schedule constraints. Minimizing rework and defects can also be classified as success criteria of the development process. The team's efficiency in conducting development operations and the effectiveness in interacting with each other and other stakeholders also denote a successful development process.

Product success corresponds to the attainment of the software's predefined *functional* and *quality* requirements. Product success differs from IS success (Delone and McLean 2003). Software functionality is assumed in Delone and McLean's IS success model whereas it is to be realized in the software project success model. In our conceptualization of software project success, software functionality can be evaluated after the development process. However, software quality elements such as response time or precision can only be effectively evaluated after the software has been implemented.

Table 2.6 Classifying Software Project Success Criteria Using the Proposed Model

Success Dimension	Criteria				References
Software development process success	- On time - On budget - Efficiency of operations	- Minimizing backlog - Minimizing bugs - Minimizing rework - Effectiveness of interaction			Ojiako et al. (2008), Baccarini (1999), Deephouse et al. (1996), Mahaney and Lederer (2011), Sauer et al. (2007); Tiwana (2012), Nidumolu (1995, 1996); Wu et al. (2008); Cooprider and Henderson (1991); Gopal and Gosain (2010); Nidumolu and Subramani (2003); Tiwana (2012), Chen et al. (2011); Ravichandran and Rai (2000); Jiang et al. (2006); Jiang et al. (2007).
Product success	***Functionality***	***Quality***			Basten et al. (2011), de Bakker et al. (2010), Chang et al. (2010), Hsu et al. (2012), Jun et al. (2011); Liu et al. (2011b); Liu et al. (2010); Liu et al. (2011c); Karlsen and Gottschalk (2004), Guimaraes et al. (2003), Deephouse et al (1996), Barki et al. (2001), Ravichandran and Rai (2000), Nidumolu (1995, 1996), Lucas et al. (1988); Gopal and Gosain (2010); Shenhar et al. (2002), Barki et al. (2001), Subramanian et al. (2007), Ravichandran and Rai (2000), Wu et al. (2008).
	- According to specification - Functional completion - Portability - Navigatability	- System quality - Performance - Reliability - Maintainability - Recoverability - Service quality - Enhanceability - Recency of information	- Information quality - Integrity - Response time - Precision - Usability - Reusability - Life expectancy - Understandability - Flexibility		

Software implementation process success	- On time - On budget - Amount of produced work - Interaction effectiveness with non-members	- Efficiency of system integration with 3rd party products or legacy systems - Efficiency of data conversion - Efficiency of operations - Quality of the client–consultant interactions	Liberatore and Lup (2010); Markus and Tanis (2000); Aladwani, (2002); Liberatore and Lup (2010); Pan et al. (2008); Häkkinen and Hilmola, (2008).
Operational success	***Productivity***	***Resource reduction***	Thong et al. (1996), Dvir et al. (2003), Irani et al. (2005), Zhu et al. (2010), Wang et al. (2007), Wixom and Watson (2001), Karlsen and Gottschalk (2004), Mahaney and Lederer (2006), Irani and Love (2000), Jahangir (2007), Palvia et al. (1992), El-Sawah et al. (2008), Zhu et al. (2010), Sasidharan et al. (2012), Karlsen and Gottschalk (2004), Atkinson (1999), Wang et al. (2007), Saarinen (1996).
	- Staff productivity - Resource utilization - Quality of decision making - Operational Efficiency of organization - Increase throughput - Reliability - Improve customer relationships - Product and service innovation	- Waste reduction - Cycle times reduction - Inventory costs reduction - Raw material inventory reduction - Manufacturing cost reduction - Labor cost reduction	

Software Assimilation Process Success	***Use***	***Temporary impact***	Liang et al. (2007), Sharma and Yetton (2003, 2007) Massetti and Zmud (1996), Naveed (1996), Yetton et al. (1999), Silva and Hirschheim (2007), Wixom and Watson (2001), Naveed (1996), Markus and Tanis (2000), Sanchez (2004), Häkkinen and Hilmola (2008).
	- Volume of use - Diversity of use - Depth of use - Frequency of use	- User resistance - Time to reach normal performance -Temporary changes in KPIs -Temporary impacts on the organization's suppliers and customers	
Business success	***Financial***	***Strategic***	Palvia et al. (1992), Dvir et al. (2003), Thong et al. (2007), Nidumolu and Subramani (2003), Barki et al. (2001), Jahangir (2007), Tsai et al. (2012), Nelson (2005), Karlsen and Gottschalk (2004), Atkinson (1999), Shenhar et al. (2002), Kearns (2007), Cats-Baril and Jelassi (1994), Irani and Love (2000), El-sawah et al. (2008), Chang et al. (2005), Thong et al. (1997), Saarinen (1996), Thong et al. (1994).
	- Return on investment - Sales revenue - Profitability	- Competitive position - Market responsiveness - Organizational learning - Market share - Commercial success - New market penetration - New product line or innovations - New opportunities - Reputation - Business threat response - Customer relationship	

The main objective of the implementation phase is to *install and configure* the software to align with the client's operational demands and business processes (Wagner and Newell 2007). Many criteria that represent the success of the development process such as being on time, on budget, and producing the planned amount of work are also criteria of the implementation process. However, other criteria are specific to the implementation process. During the implementation stage, other activities are performed that pertain to integrating the newly developed software with existing ones as well as migrating to the new information system. Accordingly, the efficiency of operations in the integration with 3rd party software or legacy systems and database as well as migrating existing databases into the new software are considered success criteria of the implementation process.

Operational success stands for *internal* organizational improvements as a result of the newly introduced software (DeLone and McLean 2003). Specifically, it relates to improvements in organizational productivity as well as reduction in required resources. It is often specified using key organizational performance indicators (Markus and Tanis 2000).

The objective of the assimilation phase is to transfer organizational functions to the newly implemented system which could interrupt the daily operations of the client organization (Liang et al., 2007). Its success is comprised of two components. The first component is the ability to swiftly move to the new system while minimizing user resistance and the temporary negative impact on current operations (Markus and Tanis 2000). The second is the volume, diversity, depth and frequency of use the assimilation achieves (Liang et al. 2007).

Business success can be specified according to the organization's financial and strategic objectives. It can only be assessed in the longer term; often years after the termination of the assimilation process. The achievement of financial and strategic objectives are increasingly viewed as important criteria of project success (e.g., Wateridge 1998). This is because internal operational success (e.g., reduced labor cost) does not necessarily translate into financial and/or strategic benefits (Ojiako et al. 2008).

A Stakeholder Perspective of Software Project Success

A stakeholder is defined as “any group or individual that can affect or is affected by the achievement of a corporation’s purpose” (Freeman 2004). This perspective highlights the need for management to include personal interests to the way they determine value (Freeman 2004). When the interests of stakeholders are in conflict, management should address the needs of different stakeholder groups (Freeman et al. 2012). By doing so, more value can be created (Freeman et al. 2012). Project stakeholders with different, and at times conflicting goals influence and are influenced by the success or failure of the software project (Sherif et al. 2006).

In the information systems literature, stakeholder satisfaction has been widely used as a surrogate of project success (e.g., Jiang and Klein 1999; Procaccino et al. 2005). The literature distinguishes stakeholder satisfaction with the product from satisfaction with the process (e.g., Baccarini 1999; Thomas and Fernández 2008; Wateridge 1998). Stakeholders could be satisfied with the process but not with its outcome and vice versa. For example, users could be dissatisfied with the communication process during development while being satisfied with the product`s quality. Similarly, IS staff could be dissatisfied because of a lack of enjoyment or learning while satisfied with the product’s quality.

Certainly, measurement instruments for different stakeholder groups are not alike. When end-user satisfaction with the project management process is assessed, items such as satisfaction with user involvement and communication quality have been employed (Jiang et al., 2001). IS-staff satisfaction are linked to learning, personal development, and work enjoyment (Pereira et al. 2008).

Satisfaction with project outcomes is also perceived differently depending on the stakeholder group. For example, end-users might be satisfied with the information system due to aspects like ease-of-use and performance. However, executives might be dissatisfied with the product for reasons like the incapability to expand its modules for future needs. Line managers might be dissatisfied because of issues like data security and portability.

However, using satisfaction as a surrogate has been criticized since people are usually unable to aggregate all their thoughts when forming their attitudes (Ostrom 1989). Hence, stakeholder satisfaction with the process could be better regarded as a distinct and more refined dimension of software project success.

Organizing the Existing Success Criteria

We examined the articles we identified to determine how different stakeholders viewed project success; specifically project managers, client and vendor top management, end users, and IS staff members. In general, the literature shows that stakeholder groups attribute different criteria in their assessment of software project success.

Project managers are more likely to perceive an on-time and on-budget project to be successful than IS staff and executives (Procaccino and Verner, 2006; Linberg, 1999). Only 62% of project managers believe that providing the best return on investment (ROI) is more important than delivering the project under budget (Amber, 2007), compared to 87% of other business stakeholders who believe that ROI is more important than delivering under budget. A similar pattern is observed in relation to delivering the product only when it is ready vs. on-schedule.

However, the IS staff and end users care the least about project cost and they identified project success based on different criteria (Nelson 2005). IS staff attributed project success to the success of the process (planning and control) (Procaccino et al., 2005), achieving the required product functionality (Agarwal and Rathod 2006; Nelson 2005), high product quality (Agarwal and Rathod, 2006; Procaccino and Verner, 2006; Nelson, 2005), learning and personal development, creativity and enjoyment (Procaccino et al., 2005), doing challenging work (Gardiner and Stewart 2000; Linberg 1999). End users viewed project success in terms of the usability of the software product (Nelson 2005).

Executive management tends to view success more in terms of the business objectives (Shenhar et al., 2001; Linberg, 1999; Witcher et al., 2007). One of the project success criteria that chief executives attributed success to is stock prices (Shenhar et al., 2002). Indeed, a study by Ojiako et al. (2008) showed that all executive managers saw project success in terms of its strategic objective. In contrast, only 40% of project managers included the project's strategic objective as a success criterion. In terms of financial objectives, Gardiner and Stewart (2000) observed that in most cases project managers did not appraise software projects as investments but as a cost.

The literature also reveals that project stakeholders not only have different success criteria but also opposing ones. One of the different success criteria stressed by Taylor (2007) is the vendor executives' high emphasis on the project's returns on investment which usually has a positive linear correlation with the client's project cost. Pertaining to the client executives, an important success criterion is minimizing project cost in order for them to realize their own return on investment. Taylor (2007) then argues that the main challenge in software projects is to balance both vendor and client organization objectives. Other conflicting success criteria can be established between client executives and end-users. Often, a top project operational objective of client executives is to reduce labor cost (Aubert et al. 2008). This criterion is in conflict with end users' project objectives of preserving their jobs (Aubert et al. 2008; Wong 2003).

In view of the above, software project success criteria identified in the previous section can be further classified according to their degree of significance for the project stakeholders. In the previous section we identified six project success dimensions of which three pertain to the success of software project processes and three apply to the success of software project outcomes. This multi-dimensional view combined with a multi-stakeholder perspective can lead to a comprehensive and clear representation software project success. Software project success criteria could be systematically organized in a multi-dimensional multi-stakeholder framework as shown in table 2.7 below.

Table 2.7 A Multi-dimensional Multi-stakeholder Software Project Success Construct

Success Dimension / Stakeholder	Development Process Success	Implementation Process Success	Product Success	Assimilation Process Success	Operational Success	Business Success
Project Manager	*Time, cost, efficiency, control, and task accuracy.*	*Time, cost, efficiency, control.*	*Functionality, quality.*	*Degree of use.*	*Personal achievement.*	*Reputation, interpersonal relationship.*
IS Staff	*Learning, creativity, personal development, satisfaction, enjoyment, doing challenging work.*	*Learning, personal development, Satisfaction, enjoyment, doing challenging work.*	*Sense of achievement of high quality.*	*A sense of delivering sufficient quality.*	*Sense of quality, job satisfaction.*	*Sense of achievement.*
Users	*Time, satisfaction with communication and involvement.*	*Time, learning, minimal business disruption, satisfaction with communication and involvement.*	*Improved quality of work (functionality and quality), ease of use.*	*Satisfaction, personal, learning, career development*	*Improved work conditions, quality of decision making*	*Job security, organizational commitment, intent to remain, performance.*
Vendor Executives	*Cost, time, organizational learning, Flexibility facing labor, market, and regulatory fluctuation.*	*Cost, time, reputation, organizational learning.*	*Market penetration, module growth, reusability.*	*Reputation, Continuing relations (guanxi)*	*Reputation, reoccurring business.*	*Financial and strategic objective, learning, reoccurring business.*
Client Executives	*Schedule and budget.*	*Schedule and budget.*	*Reusability, Longevity, Quality.*	*Temporary KPI Changes (-), resistance (-).*	*Productivity (e.g., resource utilization), resource reduction (e.g., reduced resources, waste)*	*Financial and strategic objectives*

Discussion

The proposed construct defines software project success according to the success of a sequence of processes and their associated outcomes as perceived by the project stakeholders. It helps organize the success criteria provided in information systems research according to the project's phases and their associated outcomes from the point of view of five key stakeholder groups. One advantage of combining the process and stakeholder perspectives to conceptualize project success is that it integrates the many facets of software projects in a logical way. Existing research on software project success advances the relevant knowledge by offering extensive lists of criteria that can indicate success at certain stages of the project and from the point of view of a subset of stakeholder groups. However, it is not clear how the criteria can be effectively combined to evaluate success especially when they are in conflict. A closer look at the specification of success in table 2.7 above exposes many such conflicts. These conflicts can be stakeholder related. For instance, being creative and learning which are criteria attributed to the IS staff often takes additional time that the project manager is not willing to invest. Conflicting criteria could also be time related. Short-term and long-term project objectives are not necessarily in agreement. For instance, one of the product success criteria that vendors emphasize is associated with developing reusable software components and libraries which could help in penetrating new markets in the longer term. Yet, developing generic components has two short-term consequences. First, it takes more time and effort which is in conflict with the success criteria that project managers favor. Second, it can reduce product quality and performance because of the additional programming abstractions and algorithms it requires. Besides, it conflicts with the success criteria of the project team and end-users.

There are numerous examples in which success criteria that the IS literature suggests conflict with one another. Indeed, by giving equal weights to a list of criteria without examining how they are linked can erroneously simplify the multi-faceted and intricate construct of project success. Therefore, from the proposed success construct exhibited in table 2.7 we infer that project success is more like finding the balance between a number of possibly opposing objectives. This balance is a function of the objectives of the different stakeholder groups, both in the short and long term.

Accordingly, we consider the specification demonstrated in the above table as a first step forward to a theoretically grounded software project success construct. It exposes how different project dimensions and stakeholder categories can be combined to provide a cohesive view of project success.

Although our conceptualization provides an integrated view of success, offering a way to measure it is challenging. First, determining a unit of measurement that can be used to measure criteria of different nature can be difficult. In other words, criteria like reusability, learning, quality, cost, and sense of achievement are of different natures. One approach that could be useful is to evaluate the criteria as monetary values and use net present value (NPV) as a measure of success. This approach has been suggested as a way to monetize project risk associated with aspects of project risk of different nature like user competency and client's technical knowledge (Davis 2002). Accordingly, project managers could weigh criteria they consider important in terms of dollars. For instance, if a project manager considers the satisfaction of the project team as a key criterion of success, he or she can attach a certain monetary amount to criteria like "sense of achievement" and "doing challenging work". When the project terminates, its level of success can be determined by summing the monetary values that were initially attributed to the success criteria.

When it is not possible to convert all success criteria into monetary values, a method called analytical hierarchy process (AHP) might be a more appropriate approach to evaluate success. AHP is method that can be used to decompose complex problems (project success) into a hierarchy of interrelated elements (Saaty 1980) or success criteria in the context of software projects. Accordingly, project success can be broken down into a three-level hierarchy. The highest level has a single element i.e., project success. The second level includes the success categories which are the 6 x 5 categories in table 2.7. As for the third level, the elements are the success criteria belonging to the 30 categories. Afterwards, the rank of each of the success criterion can be established using the pair-wise comparison method. In this step, the importance of each criterion is ranked with respect to the other criteria within the same category and with respect to criteria in the other categories. Lastly, the criteria weights in each level are computed using the eigenvalue method. This method has been suggested as an approach to model IT project success (Rodriguez-Repiso et al. 2007).

Conclusion

The proposed construct answers the call to move away from a view of success in terms of a limited set of criteria like cost, time, and quality towards a theoretically grounded view. Certainly, time and cost are important success indicators. However, looking at those measures in isolation provides only a curtailed representation.

The conceptualization presented in this paper offers a representation of software project success as a multi-dimensional and multi-stakeholder construct. It demonstrates how projects *progress* towards success in three software processes, each resulting in one form of project outcome. The proposed construct methodically classifies existing criteria.

Additionally, the consideration of the stakeholder perspective of software project success enriches the construct's conceptualization. This is especially true when stakeholders have conflicting objectives. For research, the inclusion of the stakeholder perspective helps explain why one stakeholder group perceives a project as a failure while another perceives it as a success. For practitioners, it provides a framework they can use to balance project objectives according to the importance of the different stakeholder groups.

References

Abdel-Hamid, K. T., Sengupta, K., and Swett, C. 1999. "The impact of goals on software project management: An experimental investigation," *MIS Quarterly* (23:4), pp 531-555.

Agarwal, N., and Rathod, U. 2006. "Defining 'success' for software projects: An exploratory revelation," *International Journal of Project Management* (24:4), pp 358-370.

Aladwani, M. A. 2002a. "An integrated performance model of information systems projects," *Journal of Management Information Systems* (19:1), pp 185-210.

Aladwani, M. A. 2002b. "IT project uncertainty, planning and success: An empirical investigation from Kuwait," *Information Technology & People* (15:3), pp 210-226.

Alberto Espinosa, J., Cummings, J. N., and Pickering, C. 2011. "Time Separation, Coordination, and Performance in Technical Teams," *Engineering Management, IEEE Transactions on* (PP:99), pp 1-13.

Alioni, D., Dulmin, R., and Mininno, V. 2007. "Risk management in ERP project introduction: A Review of the Literature," *Information & Management* (44:1), pp 547–567.

Ambler, S. 2007. "IT Project Success Rates Survey," http://www.ambysoft.com/surveys/success2007.html.

Andres, H. P., and Zmud, R. W. 2002. "A contingency approach to software project coordination," *Journal of Management Information Systems* (18:3), pp 41-70.

Armstrong, C., and Sambamurthy, V. 1999. "Information Technology Assimilation in Firms: The Influence of Senior Leadership and IT Infrastructures," *Information Systems Research* (10:4), pp 304-327.

Atsu, M. Y., Andoh-Baidoo, F. K., Osatuyi, B., and Amoako-Gyampah, K. 2010. "An Exploratory Study of The Contextual Factors That Influence Success of ICT Projects in Developing Nations: A Case Study of A Telecommunications Company in Ghana," *Journal of Information Technology Case and Application Research* (12:3), pp 56-81.

Aubert, B., Barki, H., Michel, P., and Roy, V. 2008. "A Multi-level, Multi-theory Perspective of Information Technology Implementation," *Information Systems Journal* (18), pp 45-72.

Baccarini, D. 1999. "The logical framework method for defining project success," *Project Management Journal* (30:4), pp 25-32.

Banker, R. D., Bardhan, I., and Ozer, A. 2006. "Understanding the Impact of Collaboration Software on Product Design and Development," *Information Systems Research* (17:4), pp 352-373.

Barki, H., and Hartwick, J. 2001. "Interpersonal Conflict and its Management in Information System Development," *MIS Quarterly* (25:2), pp 195-228.

Barki, H., Rivard, S., and Talbot, J. 2001. "An Integrative Contingency Model of Software Project Risk Management " *Journal of Management Information Systems* (17:4), pp 37-69.

Bartis, E., and Mitev, N. 2008. "A multiple narrative approach to information systems failure: a successful system that failed," *European Journal of Information Systems* (17:2), pp 112-124.

Basten, D., Joosten, D., and Mellis, W. 2011. "Managers' Perceptions of Information System Project Success," *The Journal of Computer Information Systems* (52:2), pp 12-21.

Chang, K.-c., Sheu, T. S., Klein, G., and Jiang, J. J. 2010. "User commitment and collaboration: Motivational antecedents and project performance," *Information and Software Technology* (52:6), p 672.

Chen, C. C., Liu, J. Y.-C., and Chen, H.-G. 2011. "Discriminative effect of user influence and user responsibility on information system development processes and project management," *Information and Software Technology* (53:2), p 149.

Cockburn, A. 2002. *Agile software development*, (1st ed.) Addison Wesley Professional.

Cooke-Davis, T. 2002. "The real success factors on projects," *International Journal of Project Management* (20:3), pp 185-190.

de Bakker, K., Boonstra, A., and Wortmann, H. 2012. "Risk managements' communicative effects influencing IT project success," *International Journal of Project Management* (30:4), p 444.

Davis, R. C. 2002. "Calculated risk: A framework for evaluating product development," *MIT Sloan Management Review* (43:4), pp 71-78.

Deephouse, C., Mukhopadhyay, T., Goldenson, D. R., and Kellner, M. I. 1996. "Software Processes and Project Performance," *Journal of Management Information Systems* (12:3), pp 187-205.

DeLone, H. W., and McLean, R. E. 2003. "The DeLone and McLean model of information systems success: a ten-year update," *Journal of Management Information Systems* (19:4), pp 9-30.

Deutsch, M. S. 1991. "An exploratory analysis relating the software project management process to project success," *Engineering Management, IEEE Transactions on* (38:4), pp 365-375.

Dvir, D., Raz, T., and Shenhar, J. A. 2003. "An empirical analysis of the relationship between project planning and project success," *International Journal of Project Management* (21:2), pp 89-95.

Espinosa, J. A., DeLone, W., and Lee, G. 2006. "Global boundaries, task processes and IS project success: a field study," *Information Technology & People* (19:4), pp 345-370.

Freeman, E. R. 2010. "Managing for Stakeholders: Trade-offs or Value Creation," *Journal of Business Ethics* (96), pp 7-9.

Freeman, E. R., Rusconi, G., Signori, S., and Strudler, A. 2012. "Stakeholder Theory(ies): Ethical Ideas and Managerial Action," *Journal of Business Ethics* (109:1), pp 1-2.

Freeman, R. E. 2004. "The Stakeholder Approach Revisited," *Zeitschrift fuer Wirtschafts- und Unternehmensethik* (5:3), pp 228-241.

Gardiner, D. P., and Stewart, K. 2000. "Revisiting the golden triangle of cost, time and quality: The role of NPV in project control, success and failure," *International Journal of Project Management* (18:4), pp 251-256.

Gopal, A., and Gosain, S. 2010. "The Role of Organizational Controls and Boundary Spanning in Software Development Outsourcing: Implications for Project Performance," *Information Systems Research* (21:4), pp 960-982.

Hahn, I., Bredillett, C., Kim, G.-M., and Taloc, M. 2012. "Agility of Project Manager in Global IS Project," *The Journal of Computer Information Systems* (53:2), pp 31-38.

Hahn, J. H., Mentz, E., and Meyer, L. 2009. "Assessment Strategies for Pair Programming," *Journal of Information Technology Education* (8), pp 273-284.

Hakkinen, L., and Hilmola, O. P. 2008. "ERP evaluation during the shakedown phase: lessons from an after-sales division," *Information Systems Journal* (18:1) Jan, pp 73-100.

Henderson, J. C., and Lee, S. 1992. "Managing I/S Design Teams: A Control Theories Perspective," *Management Science* (38:6), pp 757-777.

Hsu, J. S.-C., Lin, T.-C., Zheng, G.-T., and Hung, Y.-W. 2012. "Users as knowledge co-producers in the information system development project," *International Journal of Project Management* (30:1), pp 27-36.

Jiang, J. J., and Klein, G. 1999. "Risks to different aspects of system success," *Information & Management* (36:5), pp 263-272.

Jiang, J. J., Klein, G., and Chen, H.-G. 2001. "THe relative influence of IS project implementation policies and project leadership on eventual outcomes," *Project Management Journal* (32:3), pp 49-55.

Kearns, S. G. 2007. "How the Internal Environment impacts Information Systems Project Success: An Investigation of Exploitative and Explorative Firms," *The Journal of Computer Information Systems* (48:1), pp 63-75.

Law, C. H., and Ngai, E. W. 2007. "ERP systems adoption: An exploratory study of the organizational factors and impacts of ERP success," *Information & Management* (44:4), pp 418-432.

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), p 87.

Liang, H., Saraf, N., Hu, Q., and Xue, Y. 2007. "Assimilation of Enterprise Systems: The Effect of Institutional Pressures and the Mediating Role of Top Management," *MIS Quarterly* (31:1), pp 59-87.

Liberatore, M. J., and Luo, W. 2010. "Coordination in Consultant-Assisted IS Projects: An Agency Theory Perspective," *IEEE Transactions on Engineering Management* (57:2), p 255.

Linberg, R. K. 1999. "Software developer perceptions about software project failure: A case study," *The Journal of Systems and Software* (49:2), pp 177-192.

Liu, j., Wang, Q., and Ma, Q. 2011a. "The effects of project uncertainty and risk management on IS development project performance: A vendor perspective," *International Journal of Project Management* (29:7), p 923.

Liu, J. Y.-C., Chen, H.-G., Chen, C. C., and Sheu, T. S. 2011b. "Relationships among interpersonal conflict, requirements uncertainty, and software project performance," *International Journal of Project Management* (29:5), p 547.

Mahaney, R. C., and Lederer, A. L. 2011. "An Agency Theory Explanation of Project Success," *The Journal of Computer Information Systems* (51:4), pp 102-113.

Markus, M. L., and Tanis, C. 2000. "The Enterprise System Experience - From Adoption to Success," in *Framing the Domains of IT Management: Projecting the Future Through the Past*, Pinnaflex Educational Resources inc.: Cincinnati, Ohio, United States, pp. 173-207.

Markus, M. L., Axline, S., Petrie, D., and Tanis, C. 2000. "Learning from adopters' experiences with ERP: problems encountered and success achieved," *Journal of Information Technology* (15), pp 245-265.

Na, K., Simpson, J. T., Li, X., Singh, T., and Kim, K. 2007. "Software Development Risk and Project Performance Measurement: Evidence in Korea," *The Journal of Systems and Software* (80:4), pp 596-605.

Naveed, S. 1996. "An empirical test of the contingency approach to user participation in information systems development," *Journal of Management Information Systems* (13:1), pp 145-166.

Nelson, R. R. 2005. "Project Retrospectives: Evaluating Project Success, Failure, and Everything in Between," *MIS Quarterly* (4:3), pp 361-372.

Nicholas, J., and Hidding, G. 2010. "Management Principles Associated With IT Project Success," *International Journal of Management and Information Systems* (14:5), pp 147-156.

Nidumolu, S. R. 1995. "The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable," *Information Systems Research* (6:3), pp 191-219.

Nidumolu, S. R., and Subramani, M. 2003. "The Matrix of Control: Combining Process and Structure Approaches to Managing Software Development," *Journal of Management Information Systems* (20:3), pp 159-197.

Ojiako, U., Johansen, E., and Greenwood, D. 2008. "A qualitative re-construction of project measurement criteria," *Industrial Management + Data Systems* (108:3), pp 405-417.

Ostrom, E. 1989. "Microconstitutional Change in Multiconstitutional Political System," *Rationality and Society, New York: Cambridge University Press.* (1:1), pp 11-50.

Pant, I., and Baroudi, B. 2008. "Project management education: The human skills imperative," *International Journal of Project Management* (26:3), pp 124–128.

Pereira, J., Cerpa, N., Verner, J., Rivas, M., and Procaccino, J. D. 2008. "What do software practitioners really think about project success: A cross-cultural comparison," *The Journal of Systems and Software* (81:6), pp 897-907.

PMI, S. c. 2004. *A guide to project management (PMBOK).* (2004 ed.) Project Management Institute: Newton square, PA, USA.

Procaccino, D. J., Verner, J. M., Darter, M. E., and Amadio, W. J. 2005. "Toward predicting software development success from the perspective of practitioners: an exploratory Bayesian model," *Journal of Information Technology* (20), pp 187–200.

Purvis, R. L., Sambamurthy, V., and Zmud, R. W. 2001. "The Assimilation of Knowledge Platforms in Organizations: An Empirical Investigation," *Organization Science* (12:2), pp 117-135.

Ravichandran, T., and Arun, R. 2000. "Quality management in systems development: An organizational system perspective," *MIS Quarterly* (24:3), pp 381-415.

Ravishankar, M. N., Pan, S. L., and Leidner, D. E. 2011. "Examining the strategic alignment and implementation success of a KMS: A subculture-based multilevel analysis," *Information Systems Research* (22:1), pp 39-59.

Reich, B. H. 2007. "Managing Knowledge and Learning in IT Projects: A Conceptual Framework and Guidelines For Practice " *Project Management Journal* (38:2), pp 5-17.

Robey, D., Smith, L. A., and Vijayasarathy, L. R. 1993. "Perceptions of conflict and success in information systems development projects," *Journal of Management Information Systems* (10:1), p 123.

Rodriguez-Repiso, L., Setchi, R., and Salmeron, L. J. 2007. "Modelling IT projects success: Emerging methodologies reviewed," *Technovation* (27:10), pp 582-594.

Saarinen, T. 1990. "System Development Methodology and Project Success: An Assessment of Situational Approaches," *Information & Management* (19:3), pp 183-193.

Saaty, T. L. 1980. *The analytic hierarchy process: planning, priority setting, resources allocation*, (1st ed.) McGraw-Hill.

Sauer, C., Gemino, A., and Reich, B. H. 2007. "The Impact of Size and Volatility on IT Project Performance," *Association for Computing Machinery. Communications of the ACM* (50:11), pp 79-84.

Sharma, R., and Yetton, P. 2007. "The Contingent Effects of Training, Technical Complexity, and Task Interdependence on Successful Information Systems Implementation," *MIS Quarterly* (31:2), pp 219-238.

Shenhar, J. A., Tishler, A., Dvir, D., Lipovetsky, S., and Lechler, T. 2002. "Refining the search for project success factors: A multivariate, typological approach," *R & D Management* (32:2), pp 111-126.

Sherif, K., Zmud, R. W., and Browne, G. J. 2006. "Managing peer-to-peer conflicts in disruptive information technology innovations: the case of software reuse," *MIS Quarterly* (30:2), pp 339-356.

Taylor, H. 2007. "Outsourced IT Projects from the Vendor Perspective: Different Goals, Different Risks," *Journal of Global Information Management* (15:2), pp 1-27.

Thite, M. 2000. "Leadership styles in information technology projects," *International Journal of Project Management* (18:4), pp 235-241.

Thomas, G., and Fernández, W. 2008. "Success in IT projects: A matter of definition?," *International Journal of Project Management* (26:7), pp 733-742.

Thong, J. Y. L., Yap, C.-S., and Raman, K. S. 1996. "Top management support, external expertise and information systems implementation in small businesses," *Information Systems Research* (7:2), pp 248-267.

Tiwana, A. 2012. "Novelty-Knowledge Alignment: A Theory of Design Convergence in Systems Development," *Journal of Management Information Systems* (29:1) Summer2012, pp 15-52.

Van Bon, J., de Jong, A., and Kolthof, A. 2007. *Foundations of IT Service Management Based on ITIL V3: an introduction*, (3rd ed.) Van Haren Publishing.

Wagner, E. L., and Newell, S. 2007. "Exploring the Importance of Participation in the Post-Implementation Period of an ES Project: A Neglected Area," *Journal of the Association for Information Systems* (8:19), pp 508-524.

Wallace, L. G., Keil, M., and Rai, A. 2004. "Understanding software project risk: a cluster analysis," *Information and Management* (42:1), pp 115-125.

Wang, T. G. E., Chen, H. G. H., Jiang, J. J., and Klein, G. 2005. "Interaction quality between IS professionals and users: impacting conflict and project performance," *Journal of Information Science* (31:4), pp 273-283.

Warkentin, M., Moore, R. S., Bekkering, E., and Johnston, A. C. 2009. "Analysis of Systems Development Project Risks: An Integrative Framework," *Database for Advances in Information Systems* (40:2), pp 8-27.

Wateridge, J. 1998. "How can IS/IT projects be measured for success?," *International Journal of Project Management* (16:1), pp 59-63.

Wixom, H. B., and Watson, J. H. 2001. "An empirical investigation of the factors affecting data warehousing success," *MIS Quarterly* (25:1), pp 17-41.

Wong, B. Year. "Applying the Software Evaluation Framework 'SEF' to the Software Development Life Cycle," 2003, pp. 281-281.

Xia, W., and Lee, G. 2004. "Grasping The Complexity of IS Development Projects," *Association for Computing Machinery. Communications of the ACM* (47:5), pp 68-74.

Xu, X., Zhang, W., and Barkhi, R. 2010. "IT infrastructure capabilities and IT project success: a development team perspective," *Information Technology and Management* (11:3), pp 123-142.

Yetton, P., Martin, A., Sharma, R., and Johnston, K. 2000. "A model of information systems development project performance," *Information Systems Journal* (10:4), pp 263-289.

Appendix A: Key Articles that Address The Notion of Software Project Success

Agarwal, N., and Rathod, U. 2006. "Defining 'success' for software projects: An exploratory revelation," *International Journal of Project Management* (24:4), pp 358-370.

Aladwani, M. A. 2002. "An integrated performance model of information systems projects," *Journal of Management Information Systems* (19:1), pp 185-210.

Atkinson, R. 1999. "Project management: Cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria," *International Journal of Project Management* (17:6), pp 337-342.

Barclay, C. 2008. "Towards an integrated measurement of IS project performance: The project performance scorecard," *Information Systems Frontiers* (10:3), pp 331-345.

Nelson, R. R. 2005. "Project Retrospectives: Evaluating Project Success, Failure, and Everything in Between," *MIS Quarterly* (4:3), pp 361-372.

Nelson, R. R. 2007. "IT project management: Infamous failures, classic mistakes, and best practices," *MIS Quarterly Executive* (6:2), pp 67-78.

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

Wateridge, J. 1998. "How can IS/IT projects be measured for success?," *International Journal of Project Management* (16:1), pp 59-63.

Chapter III – Article 2

SPECIFYING SOFTWARE PROJECT RISK [2]

Mazen El-Masri
HEC Montréal
3000 Chemin-de-la-Côte-Ste-Catherine
Montréal, CANADA, H3T 2A7
Telephone number: 514 340-6000 ext. 2553
Fax Number: 514 340-6132
Mazen.el-masri@hec.ca

Suzanne Rivard
HEC Montréal
3000 Chemin-de-la-Côte-Ste-Catherine
Montréal, CANADA, H3T 2A7
Telephone number: 514 340-6493
Fax Number: 514 340-6132
Suzanne.rivard@hec.ca

[2] A preliminary version of this essay is published in the 2010 AMCIS proceedings:
El-Masri, M., and Rivard, S. 2010. "Specifying the Software Project Risk Construct," in Proceedings of the Sixteenth Americas Conference on Information Systems: Lima, Peru.

Abstract

Information systems research acknowledges a causal relationship between software project risk and project failure. However, no consensus has yet been reached with respect to what risk constitutes and how it should be specified. Existing definitions of the software project risk construct are abundant and diverse, which can lead to fragmented scientific knowledge. Our study addresses this issue by developing and providing support for a conceptualization of the software project risk construct that integrates relevant theoretical perspectives. To identify the construct's dimensions and their interrelationships, we used a method from linguistics to semantically analyze definitions and relevant descriptions of software project risk found in the information systems (IS) literature. Following this analysis, we propose a definition of software project residual risk comprising four interrelated dimensions: project traits, undesirable events, risk management practices, and expected outcomes. We provide support for the proposed conceptualization by evaluating a structural model of the construct's dimensions and their interrelationships. To evaluate the model, a case survey of 82 software project cases was conducted and the resulting dataset was analyzed using partial least squares (PLS). We conclude by discussing the proposed construct of software project residual risk from theoretical and practical perspectives.

Keywords

Software project risk, case survey, semantic analysis, risk management, risk factors

Introduction

Software project risk has long been claimed to be a major cause of project failure (Barki et al. 1993; Boehm 1991). Empirical evidence exists to support this claim, with high levels of risk being associated with undesirable results such as low software quality, delays and budget overruns (Wallace et al. 2004a). Many researchers have studied software project risk, its management and its effect on project success. An examination of this research reveals that software project risk has been conceptualized in several different ways, which is reflected in different operationalizations of the construct. For instance, some authors define risk as the probability of occurrence of undesirable outcomes and conceptualize it as the aggregation of risk factors (e.g., Jiang et al. 2002; Keil et al. 1998; Liu et al. 2010; Lu and Ma 2004; Wallace et al. 2004a). Others define risk as a function of the probability of occurrence of undesirable outcomes and the magnitude of their associated loss (e.g., Barki et al. 1993; Boehm 1991; Charette 1989; Han and Huang 2007; Lyytinen et al. 1998). Risk has also been defined as the variance from expected outcomes, which is often conceptualized as residual risk that accounts for the mitigation practices (or options) accessible to management (e.g., Benaroch et al. 2006; Clemons 1995; Davis 2002).

The existing conceptualizations of software project risk have led to diverse structural and measurement models. From a structural standpoint, risk has been modeled as a uni-dimensional construct measured with risk factors (e.g., Liu et al. 2010), as a bi-dimensional construct comprising risk factors and project outcomes (e.g., Barki et al. 1993), and as a bi-dimensional construct comprising risk factors and risk management practices (e.g., Benaroch et al. 2006). With regard to its measurement, risk factors represent a core component. Yet, the nature of risk factors varies across operationalizations. Some operationalizations limit the notion of risk factors to attributes of a software project such as project size, complexity, and team expertise (e.g., Wallace et al. 2004b; Barki et al. 1993). Others combine project attributes and possible detrimental events such as the misunderstanding of requirements and scope changes when measuring risk (e.g., Keil et al. 1998). Elements that are akin to a lack of risk management, for example, change that is not properly managed (e.g., Schmidt et al. 2001), were also included in risk measurement specification models.

Although diverse conceptualizations can deepen our understanding of a construct by allowing for a diversity of perspectives, they can also keep scientific knowledge fragmented. A well-conceptualized and validated construct is essential in order to accumulate scientific knowledge (MacKenzie et al. 2011). It can eliminate confusion regarding what it represents, allow for an adequately specified measurement model, and better reveal its relationships with other constructs (MacKenzie et al. 2011). Accordingly, in this paper we develop and provide evidence for a comprehensive and coherent software project risk construct. To do this, we adopt the approach proposed by Burton-Jones and Straub (2006) who advocate for the reconceptualization of a construct that lacks the consensus of the research community by systematically incorporating the componential structures and measurements of the extant underlying perspectives. In this study, we review 23 years of software project risk literature and conduct a semantic decompositional analysis of software project risk definitions found in this literature. The analysis allowed for the observation of four interrelated components of software project risk: risk traits, undesirable events, risk management practices, and expected outcomes. This led us to propose a conceptualization of risk as *residual risk,* with these four components as its interrelated dimensions, and define it as the conditional probability of deviation from the expected outcomes, given that all available management practices are considered. To provide evidence for the proposed risk conceptualization, we develop and evaluate a causal risk model that mirrors the risk construct and the relationships between its components. We draw on the existing operationalizations of project risk in the information systems (IS) literature to specify the risk model with the help of a panel of experts. To validate the model, we conduct a case survey of 82 software project case studies and we analyze the data with partial least squares (PLS) path modeling. The articles that describe the 82 cases are listed in Appendix L.

The paper is organized as follows: First, the existing conceptualizations of software project risk are described and the need for a conceptual clarification is emphasized. We then analyze the semantic structure of extant risk definitions and other descriptive text by adopting a technique established in linguistics in order to develop a new conceptualization of software project residual risk. Next, we provide evidence by developing and evaluating a risk model that reflects the proposed construct's multi-dimensional structure. In this section, we describe how expert judges helped to specify the measurement model using existing risk items from the IS literature. Then, we use the measurement model as a coding scheme in a case survey we conduct in order to provide a dataset to test our hypothesis. Subsequently, the data was analyzed with PLS. Finally, we discuss our results and the study's implication for research and practice.

Software Project Risk: The Need for Conceptual Clarification

Software project risk and its management have generated a fairly large body of research. In order to capture the diverse perspectives on the topic, we review 23 years of relevant publications, from 1989 to 2012, including the leading information systems journals, project management journals, and management-related journals published by the Institute of Electrical and Electronics Engineers (IEEE).[3]

Articles that examine the risk of in-house or outsourced software development and implementation projects are included in our review. We exclude articles that investigate technical risks (such as specific data security, application security, and

[3] The AIS basket of 8 journals *(MISQ, ISR, JMIS, JAIS, EJIS, JIT, JSIS, and ISJ)*, the Project Management Journal, the International Journal of Project Management, IEEE Transaction on Engineering Management, and IEEE Transactions on Software Engineering.

architectural design problems) or that pertain to the risk of projects other than software projects (e.g., construction projects, new product development projects). The search resulted in 96 articles, which are organized into three broad conceptualizations of software project risk: risk as risk factors, risk as risk exposure, and risk as variance of outcomes (Figure 3.1).

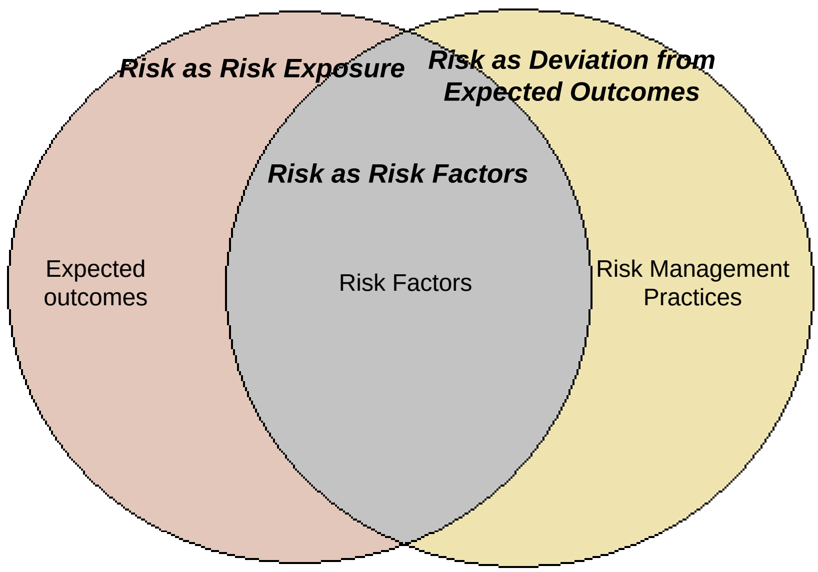


Figure 3.1 Conceptualizations of Software Project Risk in IS Literature

Risk as Risk Factors

In most of the articles (49), software project risk is defined as the probability of not attaining the expected outcomes. It is conceptualized as a set of factors – also referred to as states, properties, aspects, events, or contingencies – that can pose a serious threat to the successful completion of a software project. Examples include lack of required knowledge (Keil et al. 1998), technical complexity (Jiang and Klien 2000), user resistance (Moynihan 2002), and changing scope (Schmidt et al. 2001). Factors are frequently grouped under distinct dimensions such as team risk or requirement risk. In empirical tests, researchers often compile a list of risk factors and request that practitioners evaluate their significance (e.g., Jiang and Klein 1999).

Risk as Risk Exposure

In the second set of articles (34), software project risk is defined as risk exposure. It is conceptualized as the product of the probability of not attaining the expected outcomes and the magnitude of associated loss. This conceptualization incorporates the magnitude of associated loss as a fundamental element of risk. The premise is that while the probability of not attaining a certain expected outcome (e.g., deadline) could be low, its magnitude could be severe to project stakeholders (e.g., competitive first-to-market environment). Since the probability of undesirable outcomes is difficult to estimate (Barki et al. 1993; Heemstra and Kusters 1997), risk factor lists are often used to quantify probability distributions (e.g., Barki et al. 1993). On the other hand, magnitude of loss is defined as the significance of negative deviations from expected project outcomes. Examples include overall failure, budget overrun, delay, lower quality, user dissatisfaction, reliability, and performance shortfalls (e.g., Barki et al. 2001; Han and Huang 2007).

Risk as Variance of Outcomes

Thirteen studies define risk as deviation from expected project outcomes, which is often conceptualized as a function of risk factors as well as management risk reduction capacity. Theories of economics and strategic management are predominantly employed in these studies (e.g., real options, transaction cost, and agency theories). The deviation from project outcomes is predominantly described as negative. However, positive deviation is also considered by including risk management practices as integral components of project risk (e.g., Benaroch et al. 2006; Kumar 2002). For example, Benaroch et al. (2006) employed a real options perspective to conceptualize risk as an active component of the IS project. For these authors, the active net present value (NPV) of risk can be measured as the passive NPV (determined based on project risk factors) plus the value yielded from risk management practices. Outcomes are viewed as risk-adjusted project net present value, usually operationalized in monetary figures.

Risk Factors and the Software Project Risk Construct

The notion of risk factors is central in all three conceptualizations of software project risk. Authors who conceptualize risk as risk factors aggregate the weights of those factors in order to quantify risk (e.g., Wallace et al. 2004b). When risk is conceptualized as risk exposure, the authors use risk factors to estimate the probability of undesirable outcomes (e.g., Barki et al. 2001). Similarly, articles pertaining to the third group often use risk factors to evaluate the passive value of risk (e.g., Benaroch et al. 2006). Despite the fact that risk factors play a critical role in the specification of the project risk construct, they have two distinct definitions in the IS literature: 1) as *conditions* (attributes) that pose a serious threat to the successful completion of a software project (Wallace et al. 2004a); 2) as *contingencies* (events) that present a threat to project success (Keil et al. 1998). Yet, subscribers to the *risk factors as attributes* definition frequently use items like timetable changes, scope changes, and gold plating in their studies (e.g., Ropponen and Lyytinen 2000; Schmidt et al. 2001; Wallace et al. 2004a). Those items appear to be events that could emerge during the project rather than attributes of the project itself. However, subscribers to the *risk factors as events* definition also use project attributes like unskilled personnel and insufficient staffing (e.g., Keil et al. 1998).

Consequently, clarifying the meaning of risk factors is an essential step toward developing a well-conceptualized risk construct. According to Barki (2008), researchers must be attentive to the different “meta-categories” that a construct can comprise. Constructs that simultaneously feature behavioral, cognitive, and/or attitudinal elements often result in mixed and inconclusive research findings (Barki

2008). Distinguishing between the different metacategorical natures of a construct will not only yield more accurate models and operational definitions but also can expose the temporal interrelationships between constructs as well as between the dimensions of a single construct (Barki 2008). Pertaining to software project risk factors, time-dependent interrelationships are plausible. For example, attributes that characterize a software project at time (t_0), such as unskilled personnel, might indeed result in undesirable events like timetable changes or gold plating at time (t_1).

Reconceptualizing Software Project Risk

A clear definition is essential when an attempt is made to adequately transform a concept into a construct (Barki 2008). According to Barki's guidelines on construct conceptualization, a clear definition is the first step of construct development. Yet, the information systems literature exhibits several software project risk definitions (see Appendix A). In the social sciences, the fact that the adopted theoretical underpinnings predispose the researchers' conceptual definitions is largely accepted. However, using multiple theories with varied foundations has been shown to make up for the limitations and predispositions that are inherent in individual theories (Robey et al. 2008).

In line with Robey et al's (2008) recommendation, we seek a conceptualization of risk that incorporates the various theoretical perspectives into one all-encompassing construct. To this end, we analyzed the existing risk definitions found in related IS literature and integrated them into a single unifying definition. We then chose a methodical analytical technique called semantic decompositional analysis. We applied this technique to different definitions and relevant statements of software project risk. In linguistics, and more specifically in the subfield of semantics, text can be interpreted by the *meanings* of its words (or expressions) and the *relationships* between them (Akmajian et al. 2001). Semantic decomposition is a scientific process that aims at developing the text's semantic structure in order to reveal its meaning in a more precise manner. This is achieved by organizing the smallest units (words and expressions) of a sentence, which are called *lexical items* (e.g., "project" and "information systems"), according to their meanings. In particular, the analysis involves identification of the meaning of lexical items, the relationship between their meanings, and the relationship between the meanings of the phrases to which they belong.

In semantic decompositional analysis, sentences are made of phrases that are themselves made of lexical items (Akmajian et al. 2001). Those lexical items (i.e., words or expressions) can be identified by their *meaning properties* (Akmajian et al. 2001). Meaning properties are attributes that describe the meaning of a lexical item. There are three meaning properties (Akmajian et al. 2001, p. 237):

1. *Meaningful*: A description of the lexical item (e.g., event: an incident that happens at a given place and time).
2. *Polysemous*: A lexical item is polysemous when it has more than one meaning.
3. *Ambiguous*: A lexical item is ambiguous when its meaning is unclear in a sentence.

When meaningful, lexical items can be organized based on the *relationships* between their meanings (Akmajian et al. 2001). *Meaning relations* represent the nature of the semantic relationships between lexical items (e.g., success is the *opposite* of failure). Note that meaning relations can be assigned to a set of two or more lexical items only if the items are *meaningful*. Akmajian et al. (2001) list a number of semantic relations, three of which are relevant to our case: *synonymous* (same meaning), *antonymous* (opposite meaning), and *hyponymous* [4]. Another relevant semantic relation is a *meronymous* relationship in which a word is

[4] When the meaning of a lexical item is included in the meaning of a more general word (a *kind-of* relation).

semantically a constituent part of another word, or a *part-of* relation (Cruse 2004 p. 153).

Verb-typed lexical items are the most important in a sentence as they connect its main parts; these are the noun phrase and the verb phrase (Jackendoff 1990). Verbs can be *state* verbs (e.g., be) or *action* verbs (e.g., pose). Only the latter expose the relationships between concepts (Jackendoff 1990).

Components of Software Project Risk

In order to identify the necessary text, we started with the lexical item that we seek to define: software project risk. We searched the 96 articles for explicit definitions of software project risk. Twenty-six definitions were found (see Appendix A). Semantic decomposition was performed on these definitions. Our first step was to decompose the identified text into lexical items (see procedure in Figure 3.2). The meaningfulness of each lexical item was judged in accordance with the language and connotations that are common to software project risk and risk management literature. When items were deemed ambiguous, we searched the literature for definitions or descriptions of those items and included them in the decompositional analysis. For example, lexical items such as "chance" and "likelihood" have common meanings in project risk management and were considered unambiguous. Conversely, the item "risk factor" was polysemous because it could imply anything that could causally contribute to a result. Since clarifying the meaning of *risk factor* is imperative in our study, we searched the chosen articles for explicit definitions of the notion. In total, 11 software project risk factor definitions where included in our analysis (see Appendix B). Other ambiguous items such as "threat" and "control" were also included (see Appendix B). The decomposition analysis resulted in 92 lexical items[5], and *meaning relations* were given to them (see Figure 3.3). For example, the items *possibility, likelihood*, and *probability* have a synonymous

[5] Twenty five lexical items were action verbs.

meaning relation because they share the same meaning. The items *factor* and *condition* have a hyponymous meaning relation because a *condition* is a kind of *factor*. In addition, the items *importance* and *factor* have a meronymous meaning relation because *importance* is part of the *factor* concept.

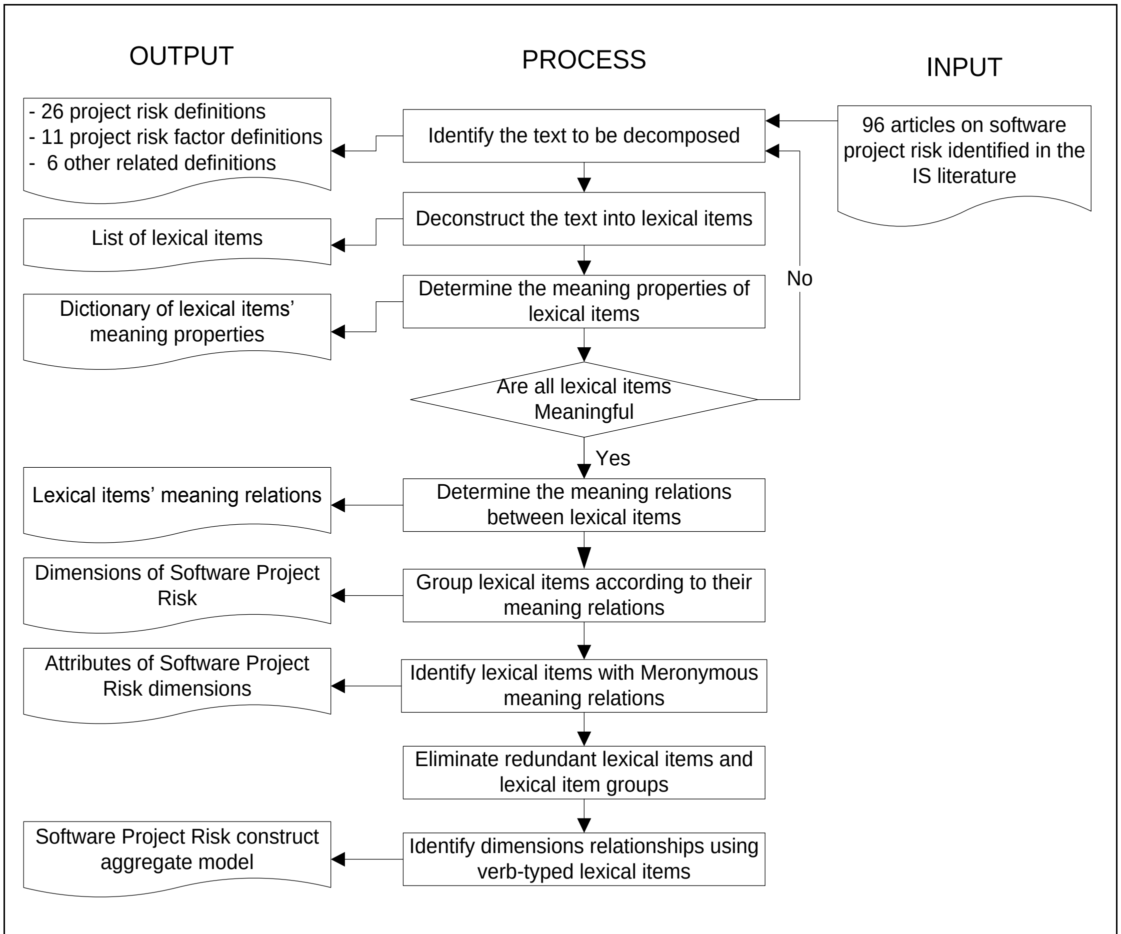


Figure 3.2 Procedure Used in Semantic Decompositional Analysis

Lexical items that share synonymous or antonymous properties were grouped together; one item judged to best describe its group was chosen to represent that group (refer to grey boxes in Figure 3.3). Attributes were then assigned to the item groups based on the meronymous (*part-of*) relations they have with other groups[6] (refer to round-edged shapes in Figure 3.3). If an item group holds hyponymous (*kind-of*) relations with another that provides more inclusive meanings, the former was considered redundant. This step in the analysis process resulted in the definition of four components of software project risk. They are risk traits, undesirable events, management practices, and project outcomes.

Risk Traits: A project risk trait is an attribute or a characteristic of the software project that represents vulnerability. It is characterized as risky based on its *relevance*, which is a value given by an evaluator (typically the project manager) quantifying the vulnerability it represents. A risk trait can be *endogenous* (e.g., team skills) or *exogenous* (e.g., highly competitive market) to the project environment. It can also be characterised by a certain *degree of control* that managers can exercise in order to reduce its relevance.

Undesirable Events: Undesirable events are probable future incidents that threaten or negatively impact the successful realization of expected outcomes. Undesirable events are characterised by the degree of control that managers can exercise to reduce the probabilities of their occurrences or their impacts on expected outcomes.

Project outcomes: Expected outcomes refer to the targeted efficiency of the project's process and the effectiveness of the information system to be completed (Barki et al. 2001; Nidumolu 1996). Outcomes can be characterised by the probability that the actual outcome would have a negative deviation from what is expected as well as the magnitude of the deviation.

Risk Management Practices: A risk management practice represents an action or an option that management can implement in order to respond to project risk. There are two types of practices:

[6] See Appendix G for examples.

1. *Mitigation practices* are managerial activities by which risk is accepted yet monitored, and a contingency plan is devised in order to recover from materialized risk.
2. *Reduction practices* are interventions that aim at reducing or eliminating project risk before it materializes.

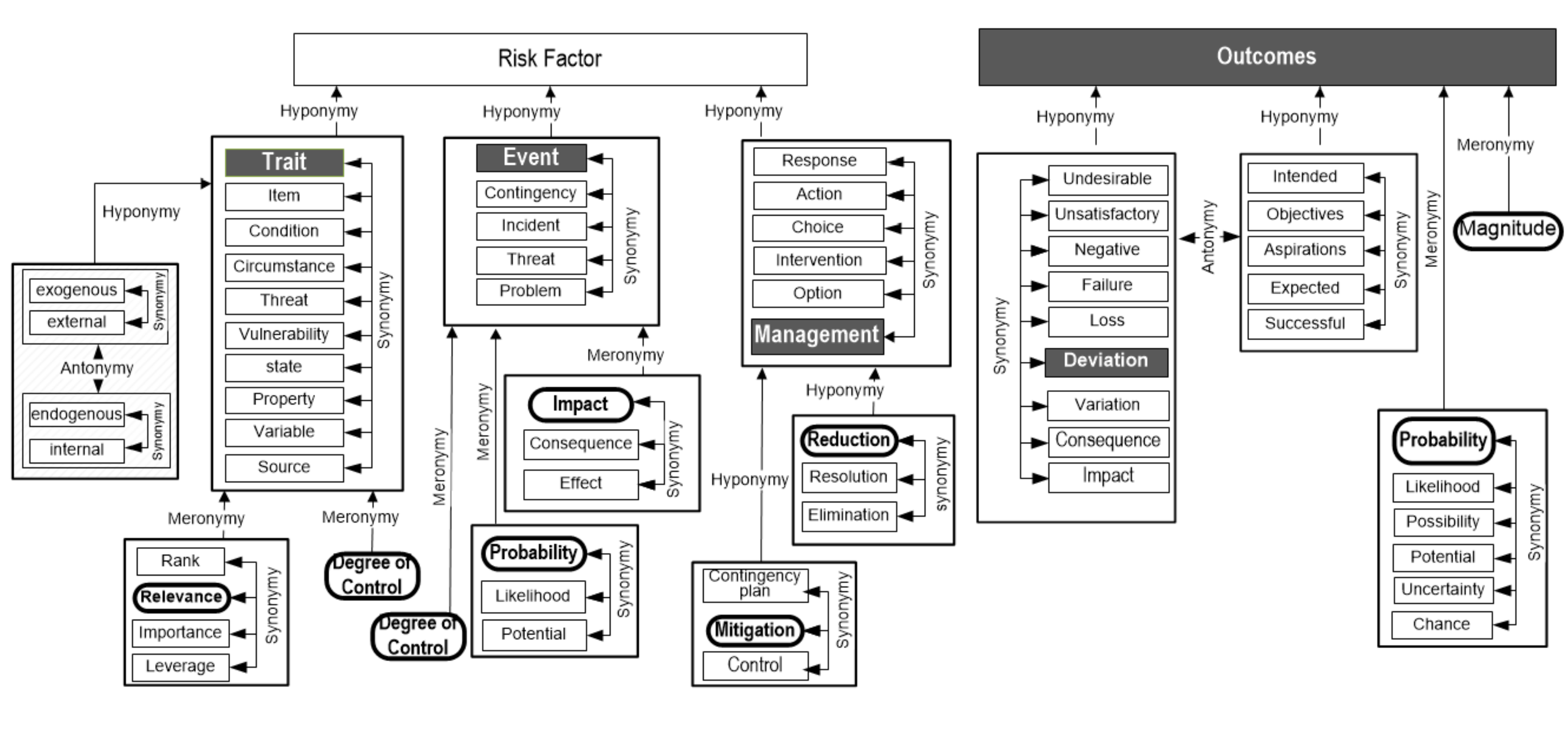


Figure 3.3 Semantic Relations of Lexical Items

Internal Structure of the Software Project Risk Construct

A software project risk construct comprising four components is a multidimensional construct. A multidimensional construct refers to "several distinct but related dimensions treated as a single theoretical concept" (Law et al. 1998). Each dimension embodies a distinct content domain of the broader construct (Polites et al. 2012). Some theoretical concepts are better specified as multidimensional constructs in order to provide holistic representations of complex phenomena and increased explained variance (Edwards 2001). Yet, it is still common in IS research to erroneously specify them as unidimensional (Petter et al. 2007), or to fail to adequately define how their dimensions relate and form the construct[7] (Wong et al. 2008).

Accordingly, a crucial step toward a well-specified multidimensional risk construct is to recognize how its four dimensions relate. The key benefit of decompositional analysis is to distinguish and relate the words (lexical items) according to their meanings and to dismiss the redundancies. The analyzed text is reduced to groups based on the items' meaning relationships.

The next step is to determine the semantic relations between the item groups in their original sentences in order to express the complete thought of the text. In other words, we need to reconstruct a new version of the text using the lexical items. The four identified risk components represent the dimensions of the software project risk construct. How those dimensions are related will expose the internal structure of a multidimensional software project risk construct. As previously stated, action verbs highlight the relationship among different concepts within a sentence (Jackendoff 1990). Accordingly, action verbs found in the original texts were used to establish

---

[7] Wong et al. (2008, p. 745) refer to constructs that group together other interrelated constructs without defining the dimensions' interrelationships and how the dimensions relate to the broader theoretical construct as "pseudo-multidimensional constructs."

how the risk dimensions relate, and, when put in a sentence, form a complete thought. There were 25 action verbs identified. Using these verbs, a version of the text was reconstructed in sentence form relating the identified lexical item groups while remaining consistent with the original texts. Figure 3.4 shows the relationships between the risk dimensions and the semantic structure of the analyzed text.

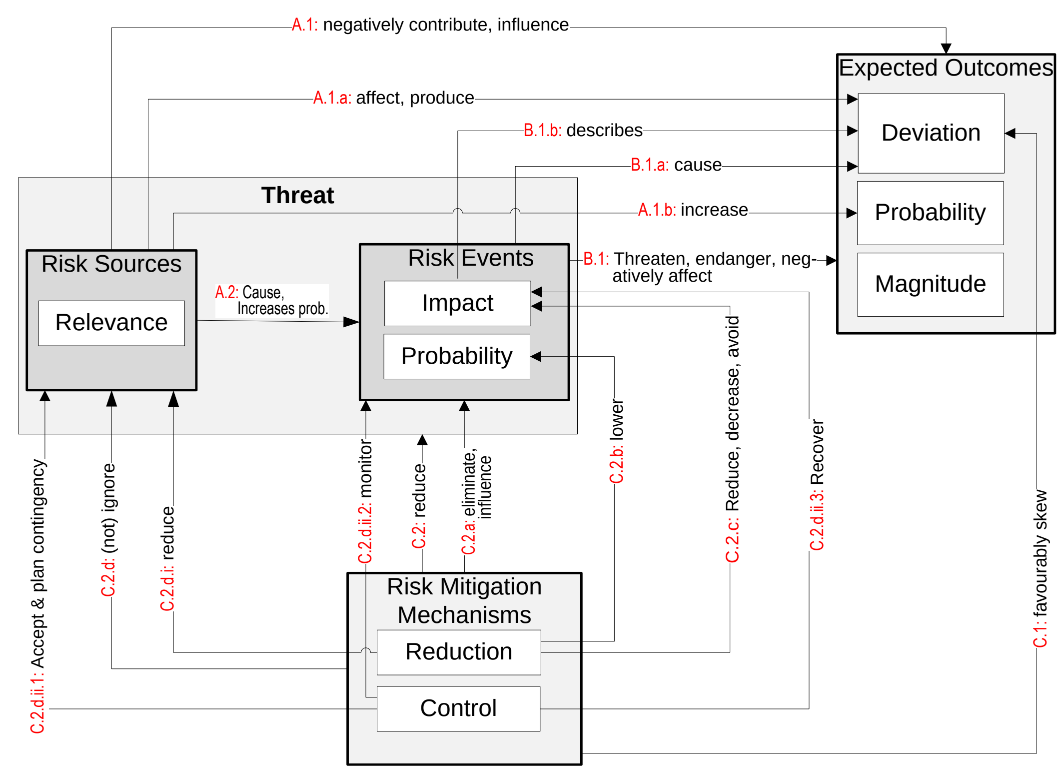


Figure 3.4 Action Verbs Lexical Items (Sentence Semantic Relations)

The reconstructed text in sentence form relates the four components of software project risk as follows:

A. A *risk trait* is associated with undesirable events and project outcomes:

  A.1. A risk trait negatively contributes to (influences) the expected project outcomes.

    A.1.a. Produces a negative variation of project outcomes.

    A.1.b. Increases the probability of negative outcomes.

  A.2. A risk trait can cause one or more undesirable events (increases the probability of occurrence of undesirable event).

B. An *undesirable event* is associated with project outcomes:

  B.1. An undesirable event negatively affects (threatens, endangers) project outcomes.

    B.1.a. It causes a variation of one or more project outcomes.

    B.1.b. The sum of the impacts of undesirable events can be described in terms of the negatively valued deviation between the intended and the actual outcomes.

C. A *risk management practice* is associated with risk traits, undesirable events, and expected outcomes. A risk management practice:

  C.1. Favourably skews variation of expected outcomes (given that its cost is less than its expected benefit).

  C.2. Reducing the threat can be achieved by:

    C.2.a. Eliminating (influencing) an undesirable event.

    C.2.b. Lowering the probability of an undesirable event.

    C.2.c. Reducing (decreasing, avoiding) the impact of an undesirable event if it occurs.

    C.2.d. Not ignoring project risk traits.

      C.2.d.i. Reducing the leverage (weight) of project risk traits.

      C.2.d.ii. Mitigate project risk traits by:

C.2.d.ii.1. Accepting the leverage (weight) of risk traits and plan contingency.

C.2.d.ii.2. Monitoring the occurrence of the undesirable event.

C.2.d.ii.3. Recovering by executing contingency plan.

A multidimensional construct is a type of construct that cannot exist separately from its dimensions (Edwards 2001), which represent its distinct and essential facets (MacKenzie et al. 2011). This type of construct exists on the same level as its dimensions and is represented by a mathematical function of their composite (Wong et al. 2008). Hence, the software project risk construct is regarded as a multidimensional construct formed by its four interrelated dimensions. Risk of deviating from expected outcomes decreases when the probability of occurrence of undesirable events decreases (associations B.1.a and B.1.b in Figure 3.4). It exists only if there is a non-zero probability that undesirable events could occur *and* that such events would impact project outcomes. Regardless of the project's risk traits and how they are associated with expected outcomes (associations A.1, A.1.a, and A.1.b in Figure 3.4), there can be no risk of deviation without the occurrence of undesirable events. Therefore, associations A.1, A.1.a, and A.1.b in Figure 3.4 cannot be a part of a risk construct that incorporates the undesirable events as a dimension. Likewise, project traits must also be a core dimension of the risk construct since the probability of the occurrence of undesirable events is a function of project risk traits. If no project traits are associated with any undesirable event, then there is no risk of deviation from planned outcomes.

Finally, the risk management practices dimension represents the active component of the construct. This dimension emphasizes the view of risk as *residual*. The risk of deviation from expected outcomes is inaccurate when the impact of management's risk reduction practices are not considered (Alter and Sherer 2004). If managers develop options that eliminate the impacts of all probable undesirable events on the expected outcomes, there is no risk. However, management practices cannot directly reduce the threat or eliminate the probability of the occurrence of undesirable events without reducing the significance of the project traits that cause them to transpire. For instance, reviewing the requirements with end-users decreases requirement ambiguity (a project risk trait), thereby reducing the probability of requirement changes. Therefore, associations C.2, C.2.a, and C.2.b are disregarded. Association C.1 is also omitted because management practices cannot directly lessen the expected negative deviation of project outcomes. The negative deviation from expected outcomes arises only when undesirable events occur. Reducing the negative impact of undesirable events on expected outcomes lessens this deviation. For instance, if all tasks are completed with no delays, there can be no deviation from the planned schedule.

According to the above analysis and consistent with the reconstructed text, we conceptualize the risk of the software project as *residual* risk and define it as the conditional probability that project risk traits will cause undesirable events (A.2) resulting in deviation from expected project outcomes (B.1.a), given that all accessible risk management practices are considered (C.2.c and C.2.d.ii.1). As discussed earlier, both the probability of deviations from expected outcomes and the relevance of risk traits are conditional on the impact of associated risk management practices. These practices reduce the relevance of project risk traits (C.2.d.ii.1) or mitigate the impact of undesirable events on project outcomes (C.2.c). Therefore:

Residual Risk = Probability [Deviation $(PO)_{1..i}$ | $UE_{1..j}$, $MP_{1..k}$], where PO, UE, and MP are project outcome, undesirable event, and mitigation practices, respectively.

The probability of occurrence of undesirable events is also conditional [$\text{Probability}\,(UE)_{1..j}$]. It can be expressed as a function of the relevance of the associated project traits given that the impact of risk reduction practices is accounted for:

$\text{Probability}(UE)_{1..j} = f(\text{Relevance}(PT_{1..l}), \text{Impacts}(RP)_{1..m})$, where PT is project trait and RP is reduction practices.

The relevance of project traits is determined by the project team [Relevance $(PT_{1..l})$]. For example, the relevance of technical complexity is set to high (2/2), medium (1/2), or low (0/2). Similarly, the project team determines the impacts of risk reduction practices [Impacts $(RP_{1..m})$]. For example, technical training will reduce the relevance of team technical incompetency from high [2/2] to medium [1/2].

While residual risk is the conditional probability of deviations from expected outcomes, residual risk exposure is that probability multiplied by the magnitude of the deviation from expected outcomes. Accordingly, we propose the following measurement model of residual risk exposure:

Residual Risk Exposure =

$$\sum_{i=1}^{n} \text{Probability}\,[\text{Deviation (PO)}_i\,|\text{UE}_{1..j}, \text{MP}_{1..k}] \times \text{Magnitude}(\text{Deviation }(PO_i))$$

The magnitude of deviation from expected outcome is determined by the project team [Magnitude (Deviation (PO_{i}))]. For example, the gravity of overspending by $10,000 is set to high [2/2], medium [1/2], or low [0/2]). The impact of risk mitigation practices is determined by the project team; reallocating or firing resistors reduces the probability of deviating from the expected level of use by 20%.

The inclusion of management practices as an active component of the proposed construct brings to light the notion of residual risk. The probability of occurrence of undesirable events is conditional on the impact of reduction practices that management performs. Likewise, the probability of deviation from expected outcomes is conditional on the impact of mitigation practices that management can implement. Therefore, to evaluate a project's risk, the options available to management that can reduce risk should be considered. A technically complex project whose manager can hire experienced consultants has a lower chance of encountering problems than an identical project that does not have this option.

Providing Evidence for The Relationships Between Components of the Software Project Residual Risk Construct

The proposed conceptualization of software project residual risk was realized by methodically analyzing existing definitions and relevant text pertaining to software project risk. In order to provide evidence of the relationships between the residual risk components delineated in the previous section, we evaluated a causal model that reflects this conceptualization. In other words, the four dimensions of risk were defined as constructs in a structural model and the relationships between them were defined as paths. To evaluate the model's structural fit, its measurement model must be specified. Data to evaluate the measurement model must also be collected and statistically analyzed. Information systems literature was examined to specify the risk measurement model. We asked project managers to organize the risk factors documented in the IS literature under the model's dimensions. A case survey was then conducted in order to collect the data required for statistical analysis. We used partial least squares (PLS) to perform the analysis and provide evidence of the model's predictive power.

Hypothesis Development

We derived a model that can provide evidence of relationships between the dimensions of the software project residual risk construct (see Figure 3.5). The four dimensions were conceptualized as follows:

1. The risk traits dimension represents the *intensity* of risky attributes of a software project.
2. The undesirable events dimension signifies the *probability* of occurrence of such events.

3. The project outcomes dimension represents the *degree of deviation* from expected outcomes.
4. The risk management practices dimension represents the *intensity* of implemented practices.

Consistent with our proposed conceptualization of residual risk and the specified relationships between its components described in the previous section, the probability that an undesirable event would materialize is a function of the intensity of the associated risk traits of the software project. Thus, our first hypothesis is:

H1: The higher the intensity of project risk traits, the more likely the project will encounter undesirable events.

The degree of deviation from expected outcomes is posited to be a function of the probabilities of the associated undesirable events. Accordingly, our second hypothesis is:

H2: The higher the intensity of undesirable events a project encounters, the more likely the project will deviate from its expected outcomes.

Risk management practices can reduce the probability of undesirable events by lessening or eliminating the relevance of risk traits. This leads to our third hypothesis:

H3: The higher the intensity of implemented risk management reduction practices, the less likely is the occurrence of undesirable events.

Risk management practices can also mitigate the impact of undesirable events and favourably skew the project's deviation from expected outcomes. Hence, our forth hypothesis is:

H4: The higher the intensity of implemented risk management mitigation practices the less likely is the deviation from expected outcomes.

The estimation of the path coefficients and the explained variances of the structural risk model as depicted in figure 3.5 provide evidence of the proposed relationships between the risk components.

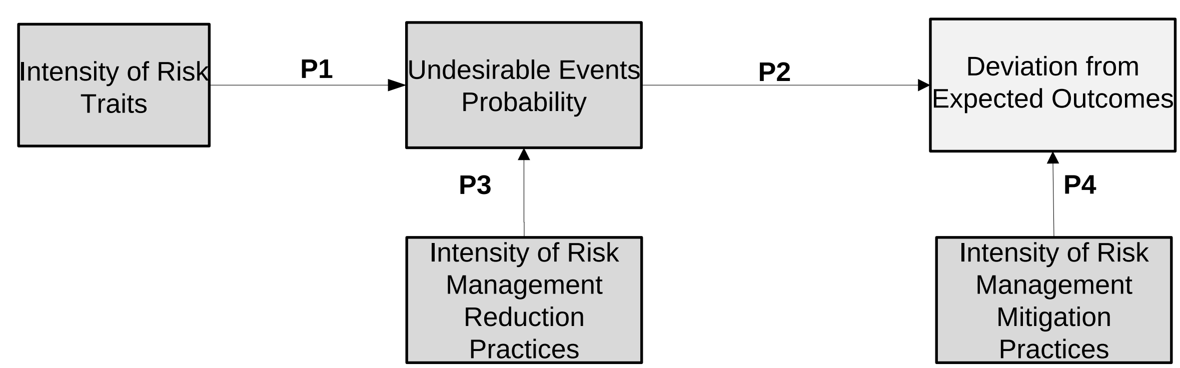


Figure 3.5 A Causal model of the Relationships between Software Project Residual Risk Components

Specifying the Measurement Model

Extant IS literature was examined to develop an operational definition for each dimension of the residual risk construct. With respect to the risk traits and undesirable events dimensions, previous IS research employs a diversity of research methods such as interviews, questionnaires and ranking-type delphi surveys [8] to develop risk factor lists. We chose articles in which authors either specifically addressed risk factor identification or empirically validated risk variables or constructs. We identified risk items from 28 articles (more details about the chosen articles can be found in Appendix D). Risk items were then analyzed in order to

[8] More details regarding the research methods employed and the associated references can be found in Appendix H, Table 1.

eliminate duplication and to consolidate the items with similar or overlapping meanings. The final list comprises 98 items. Next, this list was subjected to a series of examinations by various experts in an attempt to categorize them under either the project risk traits or undesirable events category. Card sorting was conducted to validate the content of the risk traits or undesirable events dimensions. Card sorting (or Q-sorting) is an inexpensive and reliable technique that can help realize content and construct validity (Straub et al. 2004). It requires a panel of experts, or judges, to group items under different categories (Straub et al. 2004). Typically, the researcher provides the categories along with their definitions to the judges (Moore and Benbasat 1991). Content validity is achieved when the judges assess the clarity of the wording of the different items prior to classifying them under the designated category (Moore and Benbasat 1991). The procedure also enhances convergent and discriminant validity by grouping together similar items and eliminating the ones that do not match the postulated categories (Straub et al. 2004).

The list of 98 risk items underwent three rounds of Q-sorting by experienced judges.[9] In each round, informal meetings were conducted in order to clarify disagreements regarding the risk items' categorization and to revise the wording of the risk factors. The list was first pretested by two doctoral candidates in the management of information systems. In the second round, the revised list was Q-sorted by five experienced project managers in the Information Technology and Services industry (Fleiss's kappa = 0.45). The last round of Q-sorting was conducted by three IT project managers and inter-judge agreement was reassessed using Fleiss's Kappa. A Kappa value of 0.91 was determined demonstrating a satisfactory agreement amongst the judges (Landis and Koch 1977). Our final result comprised 50 project traits and 30 undesirable events.

[9] See Appendix H for further details regarding the judges' profiles, the Q-sorting procedure followed, the intermediary inter- judge agreements, and item revisions.

The risk management practices dimension was the third dimension of software project risk that was quantified; we examined articles that empirically investigate risk management practices. This relevant research can be classified into four groups.[10] The first group includes empirical studies that apply a contingency approach to investigating the impact of aligning particular project risk profiles with risk management profiles on project success. The second group includes articles that investigate the relationship between project risk management practices and project success. The third group of studies comprises articles that investigate the relationship between certain risk management practices and project risk. The last group includes studies that empirically identify risk management practices associated with software project risk factors. In order to discover risk management practices, authors used a variety of methods such as surveys, group discussion sessions, interviews, observations, and questionnaires.

Accordingly, we made use of the articles of the fourth group to derive a set of risk management practices. The resulting list did not require validation because it comprises management practices that are often echoed in IS literature. Examples of these practices include staff training, end-user involvement, system testing, reviews, piloting, etc. Nevertheless, to ensure the comprehensiveness of our list, we contrasted the resulting risk management practices with those examined in the first three groups of articles described earlier. The risk management practices identified in those articles were analyzed and duplicates were removed, resulting in a list of 39 mechanisms. The list of 39 risk management practices is presented in Appendix F.

The fourth dimension, software project expected outcomes, has been defined as the targeted efficiency of the project's process as well as the effectiveness of the information system to be completed (Barki et al. 2001). Consistent with this definition, the prevalent IS literature on software project success conceptualizes the latter according to the success of the process and the outcome (see: Aladwani 2002a;

[10] For additional details about the four groups and selected articles see Appendix I.

Deephouse et al. 1996; Nelson 2005; Nidumolu 1995). A diversity of criteria has been suggested to capture software project process and outcomes success. However, some criteria have been validated and are used repeatedly in IS research. Adherence to the planned budget and schedule, and meeting the client's requirements/scope and quality standards, have been used consistently as the primary criteria to measure the success of the software project process (Baccarini 1999; Deephouse et al. 1996; Wallace et al. 2004a). Some authors include the amount of rework performed during the project (Ravichandran and Rai 2000), and causing minimal business disruptions (Martinez Sanchez and Perez 2004), as additional process success criteria (Ravichandran and Rai 2000). Pertaining to project outcome success, criteria that are often used are achieving the organization's objective, yielding business benefits, and level of operational use (Martinez Sanchez and Perez 2004; Nelson 2005). In keeping with the established conceptualization of software project success, we specify the expected outcomes dimension using the following nine items:

1. Adherence to planned budget;
2. Adherence to planned schedule;
3. Achieving the client's requirements/scope;
4. Meeting quality standards;
5. Amount of rework during the project (negative);
6. Degree of business disruption (negative);
7. Achieving the organization's objective;
8. Yielding business benefits;
9. Degree of use.

Data Collection Methodology

The case survey method was judged to be a suitable strategy for converting the rich IS project cases into quantifiable variables for statistical analysis. This approach is considered appropriate when: 1) there are a sufficient number of pertinent case studies in the literature, 2) the researcher is interested in the characteristics of the case rather than in the results (Yin and Heald 1975), and 3) the study phenomena are of a complex and dynamic nature (Larsson, 1998). Our research has these traits. Case survey is a research method used in order to systematically identify and statistically examine patterns across previously documented case studies (Lucas 1974; Yin and Heald 1975). The objective of case survey is to generalize by aggregating pertinent case studies into data sets that are large enough for more comprehensive statistical testing (Larsson 1993). This method takes advantage of both the data richness of qualitative case study narratives and the statistical rigor of the survey method (Larsson 1993). Hence, it transcends the lack of generalizability of case studies and the coarse-grained measures of surveying studies (Jauch et al. 1980). This is achieved via a methodical conversion of case descriptions into quantified variables and interrelations (Larsson 1993).

Case Selection Criteria

The first step in the case survey research process is to select a group of existing case studies relevant to the research objectives (Larsson 1993). Bullock (1986) stresses the importance of making the search criteria explicit when establishing the content domain. This enhances the reliability of the case survey by making the case selection process replicable (Yin and Heald 1975). Accordingly, we searched the IS literature and identified articles that include descriptions of software project cases. ABI/INFORM Global, Business Source Complete, and IEEE Xplor were the databases searched.

As is the case with traditional literature reviews, search criteria in case surveys are determined by the content domain within which hypotheses are tested (Bullock 1986). Correspondingly, our content domain broadly encompasses the entire set of case studies on IS projects. However, the quality of the case survey is contingent on the quality of the case studies selected for analysis (Larsson 1993).

Therefore, to improve the validity of the case survey, we limited the search to include articles from top journals [11] in the information systems, computer science (management), management science, and project management disciplines (see Appendix C for the list of journals searched). By tapping into top journals in closely related disciplines, we enriched our case survey. This is because case studies published in computer, project management, and organization science journals provide descriptions of IS projects from different technical, managerial, and organizational perspectives.

We established the search criteria for selecting case studies to be analyzed through numerous search tryouts and evaluations until a final set of search terms and synonyms were identified. Initially, our attempts to construct a single search criteria that would yield a complete list of IS case study articles did not work because of inconsistencies among authors in their use of common terminologies in the titles, abstracts, and associated subjects of their articles. Also, the subjects that the authors chose to associate with their articles varied greatly and were, in certain cases, missing from IS journals, preventing us as well from narrowing our result set by querying by subject. Therefore, in order to reduce any sampling bias, we performed a backward-forward case search of two IS journals. We analyzed the titles and abstracts of articles published in the last 23 years in the MIS Quarterly journal and were able to derive a set of search criteria that could yield a list of the articles that all contain case study research articles. To evaluate the soundness of the search criteria, we analyzed the titles and abstracts of articles published in the last 23 years in the European Journal of Information Systems and generated a list of case study articles. We then applied the search criteria to the same journal, and compared the two resulting lists. All case study articles identified in the former list were found in the latter, confirming the reliability of our search criteria (see Appendix C for the search criteria).

[11] The AIS senior scholar basket of eight journals, the top 50 IS journals according to Rainer and Miller's scale (2005), the top journals in management and organizational science, management journals published by the IEEE Computer Society, and the two top project management journals.

The resulting set of articles contained 1298 articles of which 295 were published in the AIS senior scholar basket of eight IS journals, 62 in organization/management science journals, 333 in IEEE journals, and 608 were published in IS journals other than the AIS basket of eight.

Inclusion Criteria

To be included, cases had to describe software development or implementation projects with the IS project as the unit of analysis. Both in-house and outsourced projects were accepted. In addition, each case had to meet the methodological rigor criteria, which is determined according to four questions:

1) Does the author include a background of the IS Project?
2) Does the author describe how the IS Project unfolded overtime?
3) Number of data collection tools used by the author(s) (e.g., interviews, documents, etc.)?
4) Number of researchers that performed the case analysis?

Exclusion Criteria

Exclusion criteria must be developed carefully in order to avoid omitting valuable cases, resulting in bias in the aggregative conclusion (Larsson 1993; Lucas 1974). They should also be stated explicitly and clearly as close-ended questions to allow for case survey replicability (Yin and Heald 1975). However, search criteria designed to identify articles with case studies must be as exhaustive as possible so that useful articles are not systematically omitted (Lucas 1974). The search criterion we designed was wide-ranging and yielded a host of articles, many of which were not case studies. Accordingly, the primary exclusion rule was whether or not the article contains case descriptions. Consequently, a large number of articles were excluded at the outset. Finally, articles that did not contain rich case descriptions also were excluded.

Sample Size

According to Gefen et al.'s (2011) rule of ten, the sample size should be at least ten times the number of structural paths in the model. Since our model has four structural paths, our minimum sample size should be no less than 40 cases. Thus, by coding 82 cases, we achieved the recommended sample size.

Coding Scheme

The primary objective of the case survey coding scheme is to employ past research findings described in the form of case studies to statistically test proposed predefined hypotheses (Bullock 1986). A coding scheme systematically converts case study descriptions into quantified variables (Larsson 1993; Lucas 1974).

Therefore, we coded the cases according to the measurement models of the four dimensions established earlier. We searched every case description for text that corresponds to the 50 project traits, 30 undesirable events, 39 management practices, and 9 project outcomes of our measurement model. As our residual risk construct suggests, risk management practices can be either reduction or mitigation practices. Therefore, we separately coded the risk management practices that were implemented prior to the occurrence of undesirable events, and the ones that were implemented after the occurrence of undesirable events. This is consistent with Alter and Ginzberg's (1978) classification, discussed earlier; they also indicated that some risk management practices can be performed as either reduction or mitigation mechanisms. In total, 82 cases were coded. See Appendix J (Section 1) for more details on the coding process.

We then used the results from the case survey to specify the dimensions of the residual risk construct. The dimensions were specified as ratio scales. Ratio variables are continuous with absolute or "nonarbitrary" zero points that possess better explanatory and predictive power than nominal or categorical variables (Bryant and

Peck 2007). Accordingly, for every case, we summed the number of codes found for each dimension and divided this value by the highest number of codes pertaining to that dimension found in all 82 cases. For instance, if the number of project risk traits identified for a specific case was three and the highest number of project risk traits identified for any case was six, then the intensity of project risk traits for the case is 0.5 (3/6). We used the same measurement technique to calculate the probability of occurrence of undesirable events, the intensity of implemented risk management reduction and mitigation practices as well as the probability of deviation from expected outcomes. Thus, the intensity of occurrence of undesirable events as well as the intensity of deviation from expected outcomes were used as proxies of their probabilities, respectively (see Section 2 of Appendix J for more details on the ratio scale specification).

Data Analysis

The proposed model was estimated by analyzing the case survey data using partial least squares path modeling (PLS-PM). PLS-PM is a component-based structural equation modeling (SEM) approach that can handle various modeling problems with greater flexibility than traditional covariance-based SEM such as AMOS or LISREL (Ringle et al. 2012; Vinzi et al. 2010). PLS path modeling is deemed more suitable than covariance-based SEM for the three reasons below. PLS path modeling was favored for regression analysis because all paths in the model can be evaluated simultaneously.

First, all the variables in our model are formative constructs that constrain our ability to estimate our measurement models (Lee et al. 2006). Moreover, the PLS statistical technique is known to be particularly beneficial during the initial theory building phase (Julien and Ramangalahy 2003). SmartPLS 2.0 M3 software was used for analyzing the data.

Evaluation of the Software Project Residual Risk Model

Component-based structural equation modeling like PLS does not provide a universal goodness of fit index such as chi-squared (Wetzels et al. 2009). Instead, inner model assessment is judged by fit indices such as the significance of the path coefficients and the coefficient of determination (R^2). Figure 3.6 presents the results from PLS statistical analysis, which shows the standardized path coefficients among the constructs using the standard PLS algorithm method and the R^2 of both undesirable events and deviation from expected outcomes. The project risk trait dimension is positively correlated with the undesirable events dimension with a path coefficient of 0.49 and variance explained $R^2 = 0.4$. The path between undesirable events and expected outcomes also is significant at 0.56 with $R^2 = 0.3$. The effect of risk reduction management practices on undesirable events is also significant at -0.36. Finally, the effect of risk mitigation management practices on project outcomes is also significant at -0.26.

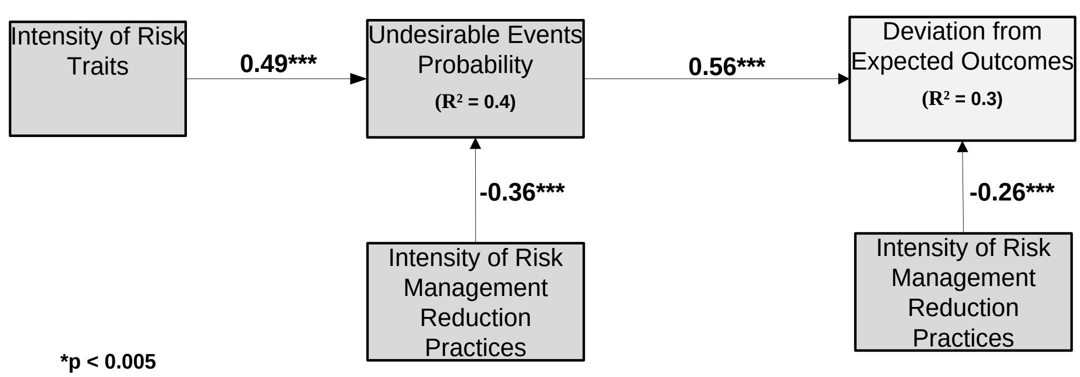


Figure 3.6 Results of the Relationships between Software Project Residual Risk Components

Discussion

Information systems research has provided valuable knowledge on the notion of risk in software projects. The various theoretical perspectives that have been adopted in this regard allowed researchers to examine software project risk from different angles. Notwithstanding the theoretical perspective researchers adopt, the notion of risk factors remain the core element in their conceptualization of risk. What differentiates the various perspectives is what risk factors represent and the way they are associated with other aspects of the software project that researchers deem relevant.

One prevailing angle we described in this paper assumes the existence of a finite set of key risk factors that can sufficiently represent software project risk. A large part of this literature restricts those factors to project traits implicitly positing that risk, which concerns possible future problems, can be conceptualized according the software project's actual state characterized by those traits. However, this view omits the myriad of possible undesirable events and management activities that can take place during a software project. Project traits pose risk to the completion of software projects only when they cause undesirable events to occur. When there is no chance of undesirable events, there is no risk (Charette 1996). Put in context, end-users' lack of experience in defining requirements (trait) can pose risk only if they take part of requirement elicitation (a management practice) and inadequately identify them (an event).

The rest of this literature indeed incorporates undesirable events alongside traits as factors that make up risk. Taking the study by Wallace et al. (2004b) as an example, requirement risk included traits related to requirement clarity and degree of conflict as well as events like their continuous changes. Yet, this part of the literature does not recognize the relationships between the traits of software project and the undesirable events. In this example, the cause and effect relation is apparent between the clarity of requirements and the possibility of them changing during the project lifetime.

The literature on software project risk has also been attentive to risk management practices, whether implicitly or explicitly. Irrespective of the theoretical perspective adopted, a considerable number of risk factor lists that we analyzed revealed that managerial actions, or lack thereof, form an important constituent of risk. These management practices implicitly took part of the researchers' conceptualization of risk. In other words, researchers view management's lack of implementation of risk management practices as risk factors. Some of these risk factors include the lack of user involvement (Al-Mashari and Al-Mudimigh 2003), poor project planning (Wallace et al. 2004b), and the lack of effective user training (Schmidt et al. 2001). However, information systems research shows that implementing risk management practices can sometimes increase risk. The excessive implementation of risk management practices for example was shown to increase risk (Ropponen and Lyytinen 2000). Additionally, risk management practices must be viewed with respect to other project traits. For example, formal planning can both reduce risk (Deephouse et al. 1996) or increase it in cases where requirements are not stable (Cone 2002).

Some of the analyzed literature was more explicit about the role of risk management practices in the conceptualization of risk. When risk is defined as the deviation from expected outcome, it was often conceptualized as the residual of risk factors and risk management practices (Benaroch et al. 2006). Yet, this part of the literature equates residual risk – risk factors minus management practices – with the deviation from expected outcomes regardless the specificities of software projects and their various intended outcomes. It does so by treating the intended outcomes as a single variable which is often attributed a monetary value. Nevertheless, software project success is an intricate concept and comprises diverse objectives like the level of use, system quality, or achieved requirements which cannot be reduced into a single variable.

The dynamics in software projects that we emphasize in this study necessitate an understanding of the risk concept that is more elaborate than the ones found in the extant perspectives. Qualitative studies that document rich description of software project cases provide a peek of the convoluted relationships between project traits, undesirable events, management activities, and project objectives. These existing perspectives have examined software project risk either by additively combining factors belonging to one or more of these risk components regardless of their inherent relationships or by considering the relationship between two out of the four components. Yet, our study demonstrates that the four fundamental components of risk are related and argues for the importance of integrating them in order to realize a holistic view of risk. Our study also provides the mathematical equations that expose the relationships between the risk components that can be used to assess risk.

A key implication of the semantic analysis we conducted is the separation of project risk factors into two distinct but related components – project traits and undesirable events – of the residual risk construct. The causal relationship between project traits and undesirable outcomes is reflected in the significant path coefficient between them (0.49) as well as the explained variance ($R^2 = 0.40$). The relationship between undesirable events and expected outcomes is also supported (Path coefficient = 0.56, $R^2 = 0.40$), lending further support to the importance of including undesirable events as a distinct dimension of the construct.

We also sought a conceptualization that synthesizes the three existing conceptualizations of risk without diverting from their associated theoretical standpoints, which led us to the notion of residual risk. The software project residual risk construct we propose coherently integrates the core risk components of the three theoretical perspectives. By incorporating risk management practices as a core component of the residual risk construct, we introduce a dynamic element to the notion of risk. This conforms to the economic theoretical perspectives of risk such as real options theory and net present value. It also answers research calls to incorporate the dynamic element of management as a fundamental element of risk (e.g., Alter and Sherer 2004; Schmidt et al. 2001).

Finally, our conceptualization of residual risk as a nonlinear combination of four interrelated components highlights the complexities associated with investigating the phenomenon. The study puts forth the relationships between risk components and provides the mathematical formulas to evaluate residual risk.

Limitations

The use of secondary data to evaluate the structural model of risk constitutes the first limitation of our study. The cases we used to conduct the case survey were intended for purposes different than that of our study. However, the rigorous inclusion and exclusion criteria that we used contribute to the soundness of our database.

A second limitation is the adoption of an indirect approach to provide evidence of the relationships between the components of the proposed residual risk construct. This was done by evaluating a causal model that reflects the construct's composition. Nevertheless, given that the empirical results well support our proposed conceptualization, we believe our efforts to maintain the meaning of each of the five constructs in the causal model, and their interrelationships, are consistent with the proposed dimensions of the residual risk construct and how they relate to one another.

Finally, the approach we used to operationalize the constructs in the causal model assumes that the items belonging to each of the five constructs have equal weight. However, this assumption may not necessarily be accurate because some items may be more important than others. For instance, the project trait "competency of IS staff" can be perceived as more relevant than the "experience of users in defining system requirements" trait. Future research that puts different weights on risk components might provide interesting insights.

Conclusion

Valuable attempts to specify the software project risk construct and establish instruments for its measurement have been presented in the IS literature. While construct specification is anchored in the researcher's approach, its merit is appraised by its predictive and explanatory power. Recently, the IS community has called for disciplined specifications of constructs because they constitute the bricks used to form all theories. Software project risk has various conceptualizations and definitions that urge the need to analyze those conceptualizations and methodically examine existing definitions.

The inspection of 23 years of pertinent literature on software project risk of different project types expands the generalizability of the construct. Semantic decompositional analysis of risk definitions and relevant text helped identify risk components and their interrelationships. As a result, the proposed conceptualization portrays risk as a multidimensional residual risk construct formed by four dimensions, namely, project risk traits, undesirable events, risk management practices, and undesirable outcomes.

Our study provides substantial advancement for research and practice. From a research perspective, the proposed residual risk construct organizes the colossal number of risk factors previously identified in the literature under four dimensions, and clarifies their relationships by explaining the ambiguity pertaining to what constitutes a risk factor. Project risk attributes belong to the project risk traits dimension; event-type factors such as requirement changes are assigned to undesirable events; managerial inaction goes under management mechanisms; and undesirable outcomes such as failing to meet user expectations fit under variations from expected outcomes. Additionally, the classification of past risk factors under the four dimensions, depending on their nature, helps eliminate redundant factors that were formerly aggregated under a single list, thus improving the accuracy of the project risk construct.

Another contribution of this research is the integration of risk management practices as a fundamental dimension of risk, leading to a conceptualization of risk as residual risk. Risk management mechanisms represent the *active* element of risk that accentuates its temporal nature. They signify the feedback-loop between future planned management mechanisms and the current state of project risk.

From a practical viewpoint, the proposed conceptualization represents a schema to better identify project risk traits at project commencement. It reduces the number of risk factors to a convenient list that managers can handle. Moreover, the proposed project risk measurement should provide managers with a more accurate assessment tool. Owing to the broad definition of software projects and scope of the literature, the construct is generalizable to all software project categories.

The proposed specification lays down a solid ground for future research pertaining to the interrelations between and within software project residual risk dimensions. Recently, empirical evidence has suggested that there exists a dynamic and complex interaction between risk components. Our well-defined software project residual risk construct is parsimonious and comprehensive, which makes it a candidate model for future research that examines the temporal interplay among its dimensions.

References

Akmajian, A., R, D., Farmer, A., and Harnish, R. 2001. *Linguistics: An Introduction to Language and Communication*, (5th ed.) MIT Press: Cambridge, MA.

Aladwani, M. A. 2002. "An integrated performance model of information systems projects," *Journal of Management Information Systems* (19:1), pp 185-210.

Alter, S., and Sherer, S., A 2004. "A General, But Readily Adaptable Model of Information System Risk," *Communications of the Association for Information Systems* (14:1), pp 1-28.

Baccarini, D. 1999. "The logical framework method for defining project success," *Project Management Journal* (30:4), pp 25-32.

Barki, H. 2008. "Thar's Gold in Them Thar Constructs," *Database for Advances in Information Systems* (39:3), pp 9-20.

Barki, H., Rivard, S., and Talbot, J. 1993. "Toward an Assessment of Software Development Risk " *Journal of Management Information Systems* (10:2), pp 203-225.

Barki, H., Rivard, S., and Talbot, J. 2001. "An Integrative Contingency Model of Software Project Risk Management " *Journal of Management Information Systems* (17:4), pp 37-69.

Benaroch, M., Lichtenstein, Y., and Robinson, K. 2006. "Real Options in Information Technology Risk Management: An Empirical Validation of Risk-Option Relationships," *MIS Quarterly* (30:4), pp 827-864.

Boehm, B. 1991. "Software Risk Management: Principles and Practices," *IEEE Software* (8:1), pp 32–41.

Bryant, C. D., and Peck, D. L. 2007. *21st century sociology: a reference handbook*, (1st ed.) SAGE Publications.

Burton-Jones, A., and Straub, D. W., Jr. 2006. "Reconceptualizing System Usage: An Approach and Empirical Test," *Information Systems Research* (17:3), pp 228-24.

Charette, R. N. 1989. *Software Engineering Risk Analysis and Management*, (2nd ed.) McGraw-Hill.

Clemons, E. K. 1995. "Using Scenario Analysis to Manage the Strategic Risks of Reengineering," *Sloan Management Review* (36:4), pp 61-71.

Cruse, D. A. 2004. *Meaning in Language: An Introduction to Semantics and Pragmatics*, (2nd ed.) Oxford University Press.

Davis, R. C. 2002. "Calculated risk: A framework for evaluating product development," *MIT Sloan Management Review* (43:4), p 71-78.

Deephouse, C., Mukhopadhyay, T., Goldenson, D. R., and Kellner, M. I. 1996. "Software Processes and Project Performance," *Journal of Management Information Systems* (12:3), pp 187-205.

Edwards, J. R. 2001. "Multidimensional constructs in organizational behavior research: An integrative analytical framework," *Organizational Research Methods* (4:2), pp 144-192.

Gefen, D., Rigdon, E. E., and Straub, D. W. 2011. "Editor's Comment: An Update and Extension to SEM Guidelines for Administrative andSocial Science Research," *MIS Quarterly* (35:2), pp iii-xiv.

Han, W. M., and Huang, S. J. 2007. "An Empirical Analysis of Risk Components and Performance on Software Projects.," *Journal of Systems and Software* (18:1), pp 42-50.

Jackendoff, R. S. 1990. *Semantic Structures*, (2nd ed.) Mit Press.

Jiang, J. J., Klein, G., and Ellis, T. S. 2002. "A measure of software development risk," *Project Management Journal* (33:3), pp 30-41.

Julien, P. A., and Ramangalahy, C. 2003. "Competitive Strategy and Performance of Exporting SMEs: An Empirical Investigation of the Impact of Their Export Information Search and Competencies," *Entrepreneurship Theory and Practice* (27:3), pp 227-245.

Keil, M., Cule, P., Lyytinen, K., and Schmidt, R. 1998. "A Framework for Identifying Software Project Risks," *Communications of the ACM* (41:11), pp 76-83.

Landis, J. R., and Koch, G. G. 1977. ""The measurement of observer agreement for categorical data," *Biometrics* (33), pp 159-174.

Larsson, R. 1993. "Case survey methodology: Quantitative analysis of patterns across case studies," *Academy of Management Journal* (36:6), p 1515.

Law, K. S., Wong, C.-S., and Mobley, W. H. 1998. "Toward a taxonomy of multidimensional constructs," *Academy of Management* (23:4), pp 741-755.

Lee, K.-H., Yang, G., and Graham, J. L. 2006. "Tension and trust in international business negotiation: American executives negotiating with Chinese executives," *Journal of International Business Studies* (37:5), pp 623–641.

Liu, S., Zhang, J., Keil, M., and Chen, T. 2010. "Comparing senior executive and project manager perceptions of IT project risk: a Chinese Delphi study," *Information Systems Journal* (20:4), pp 319-355.

Lu, X. N., and Ma, Q. G. Year. "Risk analysis in software development project with owners and contractors," Engineering Management Conference, 2004. Proceedings. 2004 IEEE International2004, pp. 789-793 Vol.782.

Lucas, W. A. 1974. "The case survey method: Aggregating case experience," Rand Corporation., Santa Monica, CA.

Lyytinen, K., Mathiassen, L., and Ropponen, J. 1998. "Attention Shaping and Software Risk: a Categorical Analysis of Four Classical Risk Management Approaches," *Information Systems Research* (9:3), pp 233–255.

MacKenzie, S. B., Podsakoff, P. M., and Podsakoff, N. P. 2011. "Construct Measurement and Validation Procedures in MIS and Behavioral Research: Integrating New and Existing Techniques," *MIS Quarterly* (35:2), p 293.

Martinez Sanchez, A., and Perez, M. P. 2004. "Early warning signals for R&D Projects: An empirical study," *Project Management Journal* (35:1), pp 11-23.

Moore, G. C., and Benbasat, I. 1991. "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *nformation Systems Research* (2:3) 1991, pp 192-222.

Nelson, R. R. 2005. "Project Retrospectives: Evaluating Project Success, Failure, and Everything in Between," *MIS Quarterly* (4:3), pp 361-372.

Nidumolu, S. R. 1995. "The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable," *Information Systems Research* (6:3), pp 191-219.

Petter, S., Straub, D., and Rai, A. 2007. "Specifying Formative Constructs in Information Systems Research," *MIS Quarterly* (31:4), p 623.

Polites, G. L., Roberts, N., and Thatcher, J. 2012. "Conceptualizing models using multidimensional constructs: a review and guidelines for their use," *European Journal of Information Systems* (21:1), pp 22-48.

Ravichandran, T., and Rai, A. 2000. "Quality management in systems development: An organizational system perspective," *MIS Quarterly* (24:3), p 381.

Ringle, C. M., Sarstedt, M., and Straub, D. W. 2012. "Editor's comments: a critical look at the use of PLS-SEM in MIS quarterly," *MIS Q.* (36:1), pp iii-xiv.

Robey, D., Im, G., and Wareham, J. D. 2008. "Theoretical Foundations of Empirical Research on Interorganizational Systems: Assessing Past Contributions and Guiding Future Directions," *Journal of the Association for Information Systems* (9:9), pp 497-518.

Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. 2001. "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems* (17:4), pp 5-36.

Straub, D., Boudreau, M.-C., and Gefen, D. 2004. "Validation Guidelines for IS Positivist Research," *Communications of AIS (CAIS)* (13:Aricle 24) April, pp 380-427.

Venkatraman, N. 1989. "The Concept Of Fit In Strategy Research: Toward Verbal And," *Academy of Management. The Academy of Management Review* (14:3), pp 423-423.

Vinzi, V. E., Chin, W. W., Henseler, J., and Wang, H. 2010. "Editorial: Perspectives on Partial Least Squares.," in *Handbook of Partial Least Squares: Concepts,* *Methods and Applications.,* S. H. o. C. S. Series (ed.), Springer Handbooks of Computational Statistics Series: Heidelberg/Dordrecht/London/New York: Springer., pp. 1-20.

Wallace, L. G., Keil, M., and Rai, A. 2004a. "How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model*," *Decision Sciences* (35:2), p 289.

Wallace, L. G., Keil, M., and Rai, A. 2004b. "Understanding software project risk: a cluster analysis," *Information and Management* (42:1), pp 115-125.

Wetzels, M., Odekerken-Schröder, G., and van Oppen, C. 2009. "Using Pls Path Modeling for Assessing Hierarchical Construct Models: Guidelines and Empirical Illustration," *MIS Quarterly* (33:1), p 177.

Wong, C.-S., Law, K. S., and Huang, G.-h. 2008. "On the Importance of Conducting Construct-Level Analysis for Multidimensional Constructs in Theory Development and Testing," *Journal of Management* (34:4), pp 744-764.

Yin, R. K., and Heald, K. A. 1975. "Using the Case Survey Method to Analyze Policy Studies," *Administrative Science Quarterly* (20:3) September, 1975, pp 371-381.

Appendix A: Definitions of Software Project Risk

References	Risk Definitions
Alberts (2006, p. 8)	The possibility of suffering loss. There are two key aspects of risk: (1) some loss must be possible and (2) there must be uncertainty associated with that loss. One additional condition is necessary for risk to be present: a choice about how to address it.
Barki et al. (1993, p. 206).	The product of project uncertainty and the magnitude of potential loss due to project failure.
Barki et al. (2001, p. 43).	The probability of an unsatisfactory outcome multiplied by the loss potential of the unsatisfactory outcome.
Bélanger and Carter (2008, p. 168)	The trustor's belief about the likelihood of gains and losses.
Benaroch et al. (2006, p. 829)	The downward or upward variation in expected outcomes.
Benaroch (2002, pp. 76-77).	The failure to respond to threats or act on opportunities.
Boehm (1991, p. 33)	The probability of an unsatisfactory outcome and the loss to the parties affected if the outcome is unsatisfactory.
Charette (1996, p. 374)	The potential for the realization of unwanted, negative consequences of an event. Risks are distinguished by two primary characteristics: their likelihood of occurrence and their potential negative consequences. Without one or the other of these present, then there is no risk.
Charette et al. (1997, p. 46).	An event with a likelihood of occurrence and some potentially negative consequence.
Cule et al. (2000, p. 2)	The negative outcome and the size of the risk is the loss incurred if the outcome should occur.
Drummond (1996, p. 347)	The chance or probability of failure.
Gluch (1994, p. 3)	The possibility of suffering a diminished level of success (loss)

	within a software-dependent development program.
Heemstra and Kusters (1996, p. 333)	The potential for realization of unwanted, negative consequences of an event. • A degree of uncertainty regarding the occurrence of the problem and • A (negative) effect on the project if the problem occurs The magnitude of the loss is referred to as risk impact. The element of uncertainty can be treated as a level of probability. Loss expectation can be expressed as the product of the risk impact multiplied by the probability and is referred to as the risk exposure.
Heemstra et al. (1997, pp. 10-11, 24)	The probability of a certain deviation between the intended and the actual output of an activity. Risk is characterised both by effect ("risk impact") as well as by likelihood ("risk probability"). `Risk impact' can be described as this negatively valued deviation between the intended and the actual outcome of the event. `Risk probability' is the likelihood that this deviation actually occurs. Risk exposure equals the sum of the (negatively valued) impact of all possible outcomes times their individual probability of occurring.
Jani (2005, pp. 11-12)	The likelihood of failure i.e. the likelihood that the project objectives will not be met. Risk can be assessed in terms of likelihood of negative outcomes and the magnitude of negative outcome.
Keil et al. (2000, p. 146)	The non-zero probability that one or more undesirable outcomes will occur; in other words, there is some likelihood of a loss.
Lauer (1996, p. 287)	Risk can be defined in terms of three components, magnitude of loss, probability of loss, and exposure to loss. Negative deviations may be considered as loss. There is uncertainty associated with each of these deviations.
Madachy (1997, p. 55).	Risk exposure is the probability of loss multiplied by the cost of the loss.
Powell and Klein (1996, p. 315)	A threat to one or more project success criteria. A standard way of describing the magnitude of risk is in terms of a probability distribution of the variable or criterion of interest. Two measures:

	the probability that the risk will occur, and the impact of the risk if it does occur.
Ropponen and Lyytinen (2000, p. 99).	A state or property of a development task or environment, which, if ignored, will increase the likelihood of project failure.
Schmidt et al. (2001, p. 7)	The product of uncertainty associated with project risk factors and the magnitude of potential loss due to project failure.
Sumner (2000, p. 317)	A problem that has not yet happened but which could cause some loss or threaten the success of your project if it did.
Wallace et al. (2004a, p. 291), Wallace et al. (2004b, p. 116).	A set of factors or conditions that can pose a serious threat to the successful completion of a software project.
Willcocks (1995, p. 7)	A range of risk factors that may contribute to negative outcomes of varying degree.
Willcocks et al. (1999, p. 286)	A negative outcome that has a known or estimated probability of occurrence
Wu (2008, p. 1033)	The expected loss attributable to failures in a project.

Software Project Risk Definitions Identified in the IS Literature

Appendix B: Definitions of Software Project Risk Factors

References	Lexical Item	Risk Factor Definitions
Alberts (2006, p. 8)	Threat	Threat is a circumstance or event that produces risk.
Alberts (2006, p. 8)	Risk Mitigation	Risk mitigation strategies are means of improving the current set of controls and thus reducing the amount of risk affecting the mission. Mitigation planning should include: (1) eliminating a triggering event, (2) monitoring for the occurrence of a trigger and implementing contingency plans when appropriate, (3) reducing vulnerability, (4) reducing potential impacts
Benaroch et al. (2006, p. 859)	Risk Control	Risk control is a strategy that accepts risk due to factors that cannot be submitted to the control of management, and it establishes contingency plans (without any further action) for recovering from materialized risk as well as possible.
Benaroch et al. (2006, p. 829)	Risk Management	Risk management is a proactive process aimed at favourably skewing the variation in expected outcomes by means of building the flexibility needed to respond to the occurrence of risk with corrective actions.
Benaroch et al. (2006, p. 858)	Risk Reduction	Risk reduction means lowering the probability of risk occurrence or its monetary consequences.
Benaroch et al. (2006, p. 829)	Risk Factor	A risk factor is a trait of an IT investment or its contextual environment that affects the degree of variation in expected outcomes. Sources of IT risk are known as risk factors.
Heemstra et al. (1997, pp. 10-11, 24)	Risk Factor	A risk source is a factor potentially causing an activity to produce a deviation between the intended and the actual output of that activity. The leverage of a risk source can be quite significant in terms of the number of different risks it may cause. A risk source refers directly to the entity which can be managed and which

		imposes one or several risks. To determine the relevance of risk sources, one should consider cause-effect type of relationships.
Jani (2005, pp. 11-12)	Risk Factor	A project risk factor is defined as a variable that can negatively influence project outcomes. Project risk characteristics can be defined by the underlying project risk factors based on the perceived managerial control over the risk factors. Project risk characteristics can be classified into endogenous and exogenous based upon the degree of managerial control over a risk factor. Endogenous risk factors are those that have the potential to affect project outcomes negatively but are under the direct control of project managers (internal to the project environment). Exogenous risk factors are those that have the potential to affect project outcomes negatively but are less under the control of project managers (external to the project environment)
Keil et al. (1998, p. 77).	Risk Factor	A risk factor is a contingency that constitutes a serious threat to the successful completion of a software development project.
Keil et al. (2002, p. 104)	Risk Factor	A risk factor is a condition that can form a serious threat to the successful completion of an IT project.
Liu et al. (2010, p. 320)	Risk Factor	A condition that can present a serious threat to the successful completion of an IT project.
Lyytinen et al. (1998, p. 236)	Risk Resolution	Risk resolution techniques are based on the espoused causal dependencies of how interventions influence risk incidents and how this will change the consequent development trajectory. Each resolution technique suggests a schematic plan for an intervention that will decrease the impact of at least one risky incident, or help avoid it all together.
Lyytinen et al. (1998,	Risk Factor	Risk items (or risk factors) are derived from postulated

p. 236)		positive causal dependency between incidents and losses.
Mursu et al. (2003, p. 182, 188)	Risk Factor	Risk factor is a contingency that can form a serious threat to the successful completion of an IS development project. Risks can be divided into the outside risks (over which the project manager has no control), and inside risks (which can be monitored and controlled). Outside risks need to be taken into the consideration in risk management, even though project managers tend to rank these risks low in importance.
Ropponen (1999)	Risk Factor	Risk factors are events, states, or actions that endanger achievement of set aspiration levels in a software development initiative.
Schmidt et al. (2001, p. 7)	Risk Factor	A risk factor is a condition that can present a serious threat to the successful completion of a software development project.

Definitions and Description of Ambiguous Lexical Items

Appendix C: Selected Journals

Category	Description	Discipline	Journals
1	Association for Information Systems senior scholars basket of eight journals	Information Systems	MIS Quarterly Information Systems Research Journal of management information systems European Journal of Information Systems Journal of the AIS Information Systems Journal Journal of Information Technology Journal of Strategic Information Systems
2	Journals rated in the top 50 Information Systems journals (excluding the AIS basket of eight)	Information Systems	Communications of the ACM Information Systems Management Information Technology & People Information and Software Technology Journal of Information Technology Case & Application Research Transforming Government: People, Process and Policy The Review of Business Information Systems Journal of Manufacturing Technology Management Journal of Organizational and End User Computing The Journal of Computer Information Systems Technology Analysis and Strategic Management Industrial Management & Data Systems Database for Advances in Information Systems Journal of Information Technology Theory and Application International Journal of Accounting Information Systems International Journal of Electronic Commerce Information Systems Frontier Sloan Management Review
3	IEEE journals oriented to the	Computer Science	IEEE Transactions on Software Engineering IEEE Transactions on Engineering Management

	management of software projects		IEEE Software
4	PM Journals	Project Management	Project Management Journal International Journal of Project Management
5	Management Journals	Organizational / Management Science	Management Science Decision Sciences Organization Science Harvard Business Review The Academy of Management Review Academy of Management Journal

Selected Journals for Case Study Selection

Search Criteria

ABS((project* OR ISD OR "Information Systems Development" OR implementation*)AND

("lesson learned" OR "lesson learnt" OR "lessons learned" OR "lessons learnt" OR case*))

OR TITLE((project* OR ISD OR "Information Systems Development" OR implementation*) AND

(experience* OR "lesson learned" OR "lesson learnt"
OR "lessons learned" OR "lessons learnt" OR stud* OR
case* OR example* OR investig* OR demonstrate* OR
analys* OR evidenc*))

AND SU[12]("Information systems" OR "Information
technology" OR "Systems development" OR "Systems
engineering" OR Software* OR Computer*)

AND PUB(<<All 38 Journal names joined with the [OR] operator>>)

[12] Only for non IS Journals

Appendix D: List of Articles that Identified Risk Factors

ID	Times Cited	Article Information	Risk Factors Identified Empirically?	Risk Variables Validated?
1	190	Title: software risk management - principles and practices Author(s): Boehm, bw IEEE Software Vol: 8 Issue: 1 pp: 32-41 Date: JAN 1991	Yes, Survey	No
2	130	Title: A framework for identifying software project risks Author(s): Keil M, Cule PE, Lyytinen K, et al. Communications Of The ACM Vol: 41 Issue: 11 pp: 76-83 Date: NOV 1998	Yes, Delphi study	No
3	118	Title: Identifying software project risks: An international Delphi study Author(s): Schmidt, R; Lyytinen, K; Keil, M, et al. Journal Of Management Information Systems Vol: 17 Issue: 4 pp: 5-36 Date: SPR 2001	Yes, Delphi study	No
4	107	Title: Toward an assessment of software development risk Author(s): Barki, H; Rivard, S; Talbot, J Journal of Management Information Systems Vol: 10 Issue: 2 pp: 209-226 Date: Fall 1993	No, Literature Review	Yes, Questionnaire
5	84	Title: Risk factors in enterprise-wide/ERP projects Author(s): Sumner, M	Yes, Interviews	No

		Journal of information technology vol: 15 issue: 4 pp: 317-327 Date: DEC 2000		
6	76	Title: Attention shaping and software risk - A categorical analysis of four classical risk management approaches Author(s): Lyytinen, K; Mathiassen, L; Ropponen, J Information Systems Research Vol: 9 Issue: 3 pp: 233-255 Date: SEP 1998	No, Literature Review	No
7	67	Title: Components of software development risk: How to address them? A project manager survey Author(s): Ropponen, J; Lyytinen, K IEEE Transactions on Software Engineering Vol: 26 Issue: 2 pp: 98-112 Date: FEB 2000	No, Literature Review	Yes, Survey
8	65	Title: An integrative contingency model of software project risk management Author(s): Barki, H; Rivard, S; Talbot, J Journal of Management Information Systems Vol:17 Issue:4 pp: 37-69 Date: Spr 2001	No, Literature Review	Yes, Questionnaire
9	58	Title: Managing Uncertainty In Mis Implementation Author(s): Alter, S; Ginzberg, M Sloan Management Review Vol: 20 Issue: 1 pp: 23-31 Date: 1978	Yes, Interviews	No
10	56	Title: Managing risks in enterprise systems implementations Author(s): Scott JE, Vessey I Communications Of The ACM Vol: 45 Issue: 4 pp: 74-81 Date: APR 2002	No, Literature Review	No

11	54	Title: Critical issues in abandoned information systems development projects Author(s): Ewusi-Mensah, K Communications Of The ACM Vol: 40 Issue: 9 pp: 74-80 Date: Sep 1997	Yes, Documentation, Case Studies	No
12	54	Title: Understanding software project risk: a cluster analysis Author(s): Wallace, L; Keil, M; Rai, A Information & Management Vol: 42 Issue: 1 pp: 115-125 Date: Dec 2004	No, Literature Review	Yes, survey
13	41	Title: How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model Author(s): Wallace, L; Keil, M; Rai, A Decision Sciences Vol: 35 Issue: 2 pp: 289-321 Date: SPR 2004	No, Literature Review	Yes, Survey
14	36	Title: Software project risks and their effect on outcomes Author(s): Wallace L, Keil M Communications Of The ACM Vol: 47 Issue: 4 pp: 68-+ Date: Apr 2004	Yes, Delphi Study	No
15	32	Title: Software development risks to project effectiveness Author(s): Jiang, J; Klein, G Journal Of Systems And Software Vol: 52 Issue: 1 pp: 3-10 Date: May 15 2000	No, Literature review	Yes, survey
16	26	Title: Risks to different aspects of system success Author(s): Jiang, JJ; Klein, G Information & Management Vol: 36 Issue: 5 pp: 263-272 Date: Nov 1999	No, Literature review	Yes, survey
17	25	Title: An inventory of personal constructs for information systems project risk researchers Author(s): Moynihan, T Journal Of Information Technology Vol: 11 Issue: 4 pp: 359-371 Date: Dec	Yes, Interviews (personal construct elicitation Technique)	No

		1996		
18	23	Title: How experienced project managers assess risk Author(s): Moynihan, T IEEE Software Vol: 14 Issue: 3 pp: 35-41 Date: MAY-JUN 1997	Yes, Interviews (personal construct elicitation Technique)	No
19	22	Title: Reconciling user and project manager perceptions of IT project risk: a Delphi study Author(s): Keil M, Tiwana A, Bush A Information Systems Journal Vol: 12 Issue: 2 pp: 103-119 Date: APR 2002	Yes, Delphi study	No
20	22	Title: Can software risk management improve system development: An exploratory study Author(s): Ropponen, J; Lyytinen, K European Journal Of Information Systems Vol: 6 Issue: 1 pp: 41-50 Date: Mar 1997	No, Literature review	Yes, survey
21	21	Title: An empirical analysis of risk components and performance on software projects Author(s): Han, WM; Huang, SJ Journal Of Systems And Software Vol: 80 Issue: 1 pp: 42-50 Date: JAN 2007	No, Literature review	Yes, survey
22	21	Title: Risk management in ERP project introduction: Review of the literature Author(s): Aloini, D; Dulmin, R; Mininno, V Information & Management Vol: 44 Issue: 6 pp: 547-567 Date: SEP 2007	No, Literature Review	No

23	21	Title: Assessing risk in ERP projects: identify and prioritize the factors Author(s): Huang SM, Chang IC, Li SH, et al. Industrial Management & Data Systems Vol: 104 Issue: 8-9 pp: 681-688 Date: 2004	Yes, Delphi study	No
24	15	Title: Real options in information technology risk management: An empirical validation of risk-option relationships Author(s): Benaroch, M; Lichtenstein, Y; Robinson, K MIS Quarterly Vol: 30 Issue: 4 pp: 827-864 Date: DEC 2006	No, Literature review	Yes, documentation
25	12	Title: Information system success as impacted by risks and development strategies Author(s): Jiang, JJ; Klein, G; Discenza, R IEEE Transactions On Engineering Management Vol: 48 Issue: 1 pp: 46-55 Date: Feb 2001	No, Literature review	Yes, survey
26	9	Title: Software development risk and project performance measurement: Evidence in Korea Author(s): Na KS, Simpson JT, Li XT, et al. Journal of Systems and Software Vol: 80 Issue: 4 Special Issue pp: 596-605 Date: Apr 2007	No, Literature review	Yes, Questionnaire
27	8	Title: Identifying software project risks in Nigeria: an International Comparative Study Author(s): Mursu A, Lyytinen K, Soriyan HA, et al. European Journal Of Information Systems Vol: 12 Issue: 3 pp: 182-194 Date: Sep 2003	Yes, Delphi study	No

28	5	Title: Exploring the relationship between software project duration and risk exposure: A cluster analysis Author(s): Huang SJ, Han WM Information & Management Vol: 45 Issue: 3 pp: 175-182 Date: Apr 2008	Yes, Delphi study	Yes, Survey

Appendix E: Card Sorting – Part 1

<u>OVERALL DESCRIPTION:</u> The objective of this exercise is to organize a list of items about a software project (e.g., the development of an information system, the implementation of an ERP) according to their *nature*. We believe that the items listed below could be organized under two different categories: project traits and events.

<u>Project traits</u> are attributes that contribute to describe a software project, and define its nature.

Events are <u>incidents</u> encountered during a software project.

<u>INSTRUCTIONS</u>

After reading carefully **the definitions** of the two categories of items, please proceed to the table below. The table lists 88 items. For each item, check the checkbox that corresponds to the category to which you believe the item belongs (in column 3). Upon analyzing the items, do not guess the category. It is highly recommended that you go back to the category <u>definition</u> in order to make sure that you are selecting the suitable category. If you cannot identify the category that the item belongs to, please <u>do not select a category</u>. If you select the category you believe the item belongs to but you are less confident of your selection check the "less certain" checkbox. You can write comments or report ambiguities relating to the wording or the meaning of the items in the comments field (column 5). You can also use this space to explain why you were less sure where the item belongs. If you need extra space, you can add your comments at the end of the document. In this case, please make sure you enter the ID of the item you are referring to (given in column 1).

Thank you for your participation. If you have any questions, don't hesitate to contact me.

Mazen El Masri

ID	Risk Item Name	Category		Confidence Level	Comments
1	Level of top management support	Trait [ ]	Event [ ]	Less certain [ ]	
2	Loss of top management support	Trait [ ]	Event [ ]	Less certain [ ]	
3	End-users' Resistance (refusing to participate, cooperate, provide requirements and/or perform acceptance testing)	Trait [ ]	Event [ ]	Less certain [ ]	
4	Degree of sophistication (expectations) of end-users	Trait [ ]	Event [ ]	Less certain [ ]	
5	Experience of end-users' representatives to define system requirements	Trait [ ]	Event [ ]	Less certain [ ]	
6	Knowledge of end-users' representatives with the application domain	Trait [ ]	Event [ ]	Less certain [ ]	
7	Level of end-users commitment to the project	Trait [ ]	Event [ ]	Less certain [ ]	
8	IT competence of customer organization	Trait [ ]	Event [ ]	Less certain [ ]	
9	Sufficiency of required budgetary resources to conduct the project	Trait [ ]	Event [ ]	Less certain [ ]	
10	Sufficiency of required time require (person hours) to conduct the project	Trait [ ]	Event [ ]	Less certain [ ]	
11	Sufficiency of human resources to conduct the project	Trait [ ]	Event [ ]	Less certain [ ]	
12	Shortfall encountered in IT Personnel performance/efficiency	Trait [ ]	Event [ ]	Less certain [ ]	
13	Extent of customer's responsibility, ownership, and buy-in of the project and its delivered system(s)	Trait [ ]	Event [ ]	Less certain [ ]	

14	IT personnel's knowledge of the user departments and operations	Trait [ ]	Event [ ]	Less certain [ ]	
15	IT personnel's overall administrative skills	Trait [ ]	Event [ ]	Less certain [ ]	
16	IT personnel's expertise with the information system's business domain	Trait [ ]	Event [ ]	Less certain [ ]	
17	IT personnel's development/ implementation expertise (methodology, support tools, project management tools, and implementation tools)	Trait [ ]	Event [ ]	Less certain [ ]	
18	Inadequate design (e.g., performance shortfalls)	Trait [ ]	Event [ ]	Less certain [ ]	
19	Clarity of system requirements	Trait [ ]	Event [ ]	Less certain [ ]	
20	Availability of system requirements	Trait [ ]	Event [ ]	Less certain [ ]	
21	Degree of stability of system requirements	Trait [ ]	Event [ ]	Less certain [ ]	
22	Changes occurred to system requirements or scope	Trait [ ]	Event [ ]	Less certain [ ]	
23	Clarity of role definitions of team members	Trait [ ]	Event [ ]	Less certain [ ]	
24	Conflict encountered between project stakeholders	Trait [ ]	Event [ ]	Less certain [ ]	
25	Degree of conflict within system requirements of various stakeholders' groups (level of agreement/ differences in project goals, deliverables, design, etc.)	Trait [ ]	Event [ ]	Less certain [ ]	
26	Inadequately identifying (misunderstanding) system requirements	Trait [ ]	Event [ ]	Less certain [ ]	

27	Degree of maturity of used technology	Trait [ ]	Event [ ]	Less certain [ ]	
28	Degree of stability of the business and organizational environments	Trait [ ]	Event [ ]	Less certain [ ]	
29	Degree of alignment between organizational culture and the business process changes required by the new IS.	Trait [ ]	Event [ ]	Less certain [ ]	
30	Changes in ownership or senior management during project	Trait [ ]	Event [ ]	Less certain [ ]	
31	Availability of end-users' representatives to support the IT team	Trait [ ]	Event [ ]	Less certain [ ]	
32	Diversity of stakeholder groups (and end-user organizational units)	Trait [ ]	Event [ ]	Less certain [ ]	
33	Project manager's expertise	Trait [ ]	Event [ ]	Less certain [ ]	
34	Level of institutionalization of effective project management methodology, structure, and standards	Trait [ ]	Event [ ]	Less certain [ ]	
35	Scope creep (uncontrolled changes to the scope)	Trait [ ]	Event [ ]	Less certain [ ]	
36	Gold-plating (adding unnecessary features due to professional interest or pride or user demands)	Trait [ ]	Event [ ]	Less certain [ ]	
37	Soundness of the project's business case	Trait [ ]	Event [ ]	Less certain [ ]	
38	Failure to adequately estimate of scope of work (schedule, cost, human resources)	Trait [ ]	Event [ ]	Less certain [ ]	
39	Project requires the acquisition and installation of new hardware	Trait [ ]	Event [ ]	Less certain [ ]	

40	Project requires the acquisition and installation of new software	Trait [ ]	Event [ ]	Less certain [ ]	
41	The number of external subcontractors	Trait [ ]	Event [ ]	Less certain [ ]	
42	The number of external consultants	Trait [ ]	Event [ ]	Less certain [ ]	
43	Turnover rate of IT personnel	Trait [ ]	Event [ ]	Less certain [ ]	
44	Strains encountered in team relationships and collaboration	Trait [ ]	Event [ ]	Less certain [ ]	
45	Degree of stability and appropriateness of the technological architecture, infrastructure, and networks	Trait [ ]	Event [ ]	Less certain [ ]	
46	Failure of project consultants or external vendors to meet the project's external dependencies	Trait [ ]	Event [ ]	Less certain [ ]	
47	Number of external vendors furnishing software or hardware components required for the project	Trait [ ]	Event [ ]	Less certain [ ]	
48	Level of institutionalization of effective project control mechanisms over consultants, vendors, and subcontractors	Trait [ ]	Event [ ]	Less certain [ ]	
49	Availability of documentation (e.g., regarding implemented system or related legacy systems)	Trait [ ]	Event [ ]	Less certain [ ]	
50	Level of ambiguity of documentation (e.g., regarding implemented system or related legacy systems)	Trait [ ]	Event [ ]	Less certain [ ]	
51	Existence of a project champion	Trait [ ]	Event [ ]	Less certain [ ]	
52	Failure to redesign business processes	Trait [ ]	Event [ ]	Less certain [ ]	

53	Number of hardware and/or software suppliers involved in the project	Trait [ ]	Event [ ]	Less certain [ ]	
54	Failure to integrate/interface new IS with legacy applications and other applications	Trait [ ]	Event [ ]	Less certain [ ]	
55	Resignation(s) of key project member(s)	Trait [ ]	Event [ ]	Less certain [ ]	
56	Reallocation of project members	Trait [ ]	Event [ ]	Less certain [ ]	
57	Project size (cost, time, staffing level, number of affected parties, number of end-users, and number of hierarchical levels occupied by system users)	Trait [ ]	Event [ ]	Less certain [ ]	
58	Failure to specify performance requirements	Trait [ ]	Event [ ]	Less certain [ ]	
59	Changes in project's time table	Trait [ ]	Event [ ]	Less certain [ ]	
60	Escalating project cost (going over planned budget)	Trait [ ]	Event [ ]	Less certain [ ]	
61	Escalating project time of completion (delay)	Trait [ ]	Event [ ]	Less certain [ ]	
62	Morale and level of commitment of project team members	Trait [ ]	Event [ ]	Less certain [ ]	
63	Changes in organizational management during the project (e.g., restructuring)	Trait [ ]	Event [ ]	Less certain [ ]	
64	Changes occurred in organizational priorities	Trait [ ]	Event [ ]	Less certain [ ]	
65	The degree of change the system brings (to procedures, workflow, structures, and so on)	Trait [ ]	Event [ ]	Less certain [ ]	

66	IT Personnel's general expertise (work with management, team, and effectively perform tasks)	Trait [ ]	Event [ ]	Less certain [ ]	
67	Number of links to (integration with or interfacing) existing and future systems	Trait [ ]	Event [ ]	Less certain [ ]	
68	Degree of technical complexity	Trait [ ]	Event [ ]	Less certain [ ]	
69	Degree of application complexity	Trait [ ]	Event [ ]	Less certain [ ]	
70	Level of reliability of target (customer) computer machinery	Trait [ ]	Event [ ]	Less certain [ ]	
71	Level of reliability of development computer machinery	Trait [ ]	Event [ ]	Less certain [ ]	
72	Level of coordination complexity (need to share resources, need to subcontract, and so on)	Trait [ ]	Event [ ]	Less certain [ ]	
73	Developing the wrong functions and properties	Trait [ ]	Event [ ]	Less certain [ ]	
74	Developing the wrong user interface	Trait [ ]	Event [ ]	Less certain [ ]	
75	Feasibility of technological solution	Trait [ ]	Event [ ]	Less certain [ ]	
76	Developer's knowledge of customer's country, culture, and language	Trait [ ]	Event [ ]	Less certain [ ]	
77	Criticality of the new system roll-out (ease of reverse to prior system of operations)	Trait [ ]	Event [ ]	Less certain [ ]	
78	Failure to select the appropriate packaged solutions	Trait [ ]	Event [ ]	Less certain [ ]	
79	Communication failure between the different project stakeholder groups	Trait [ ]	Event [ ]	Less certain [ ]	

80	Complex architecture and high number of implementation modules	Trait [ ]	Event [ ]	Less certain [ ]	
81	Failure to transition and migrate data from legacy system	Trait [ ]	Event [ ]	Less certain [ ]	
82	Clarity of expected investment benefits	Trait [ ]	Event [ ]	Less certain [ ]	
83	Technical problems	Trait [ ]	Event [ ]	Less certain [ ]	
84	Level of Information System's compliancy with the customer's technical architecture	Trait [ ]	Event [ ]	Less certain [ ]	
85	Compatibility of the composition of the project team with project structure (in regards to coordination, controls, etc.)	Trait [ ]	Event [ ]	Less certain [ ]	
86	Failure of the affected business units to handle the extent of change	Trait [ ]	Event [ ]	Less certain [ ]	
87	Degree of alignment between adopted IS functionality and the needs of the adopting organization	Trait [ ]	Event [ ]	Less certain [ ]	
88	Expertise of project's suppliers with the required activities	Trait [ ]	Event [ ]	Less certain [ ]	

Appendix F: Articles that Identified Risk Management Practices

No	Risk Management Mechanism	References
1	Detailed cost and schedule estimation	Boehm (1991), Tesch et al. (2007).
2	Provide training programs (including cross-training)	Alter and Ginzberg (1978), Benaroch et al., (2006), Boehm (1991), Tesch et al. (2007).
3	Check references of external suppliers or consultants	Boehm (1991).
4	Analyze project mission	Boehm (1991), Tesch et al. (2007).
5	Analyze client organization	Boehm (1991)
6	Inspect externally furnished components	Boehm (1991).
7	Conduct formal user specification approval	McFarlan (1975), Tesch et al. (2007).
8	Analyze/test product (Quality Assurance and Performance Testing)	Boehm (1991), Ben-David and Raz (2001), Benaroch et al. (2006).
9	Analyze costs and benefits	Boehm (1991)
10	Conduct progress reviews	McFarlan (1975), Tesch et al. (2007).
11	Conduct project milestone phases review	Benaroch et al. (2006), Boehm (1991), McFarlan (1975), Tesch et al. (2007)
12	Designate a project sponsor	Tesch et al. (2007)
13	Hire experienced personnel	Boehm (1991), Ben-David and Raz (2001), Tesch et al. (2007), McFarlan (1975)
14	Match tasks to personnel skills	Boehm (1991), Tesch et al. (2007)
15	Identify task dependencies (using PERT, critical path, networking...)	McFarlan (1975)
16	Perform formal change management disciplines	McFarlan (1975), Tesch et al. (2007)
17	Pilot system	Benaroch et al. (2006), Tesch et al. (2007), Boehm (1991)
18	Divide project tasks into smaller chunks	McFarlan (1975), Tesch et al. (2007)
19	Scale down/scrub requirements	Benaroch et al. (2006), Alter and Ginzberg (1978), Ben-David and Raz (2001), Boehm (1991), Tesch et al. (2007)
20	Select team members with significant previous work relationships	McFarlan (1975)

21	Hide complexity from end-users	Alter and Ginzberg (1978)
22	Avoid requirement changes	Alter and Ginzberg (1978), Boehm (1991)
23	Hire external consultants or consulting company	Ben-David and Raz (2001), McFarlan (1975)
24	Outsource	Benaroch et al. (2006)
25	Use an information hiding approach	Boehm (1991)
26	Involve end-users by: • Selecting a user as project manager • Create a user steering committee • Allow users to manage the change process • distribute team minutes to key users • Select users as team members • Make users responsible for education and installation of system • Allow users to manage decisions on key action dates	McFarlan (1975), Tesch et al. (2007)
27	Develop early users' manuals	Boehm (1991)
28	Meet frequently with team	McFarlan (1975)
29	Meet frequently with steering committee	McFarlan (1975), Tesch et al. (2007)
30	Regularly prepare and distribute meeting minutes on key design decisions	McFarlan (1975)
31	Permit voluntary use	Alter and Ginzberg (1978)
32	Insist on mandatory use	Alter and Ginzberg (1978)
33	Follow systems specification standards	McFarlan (1975)
34	Devise award-fee contracts	Boehm (1991)
35	Use Incremental approach	Boehm (1991), Benaroch et al. (2006), Ben-David and Raz (2001), Tesch et al. (2007)
36	Use evolutionary approach	Alter and Ginzberg (1978),
37	Use modular approach	Alter and Ginzberg (1978),
38	Prototype/develop scenarios	Boehm (1991), Alter and Ginzberg (1978), Benaroch et al. (2006)
39	Reuse existing software	Boehm (1991)

Appendix G: Description of the Semantic Decompositional Analysis that was Conducted

Lexical items that share synonymous properties were then grouped together. Only the item that is judged to best describe its group is chosen to represent it. For example, the items *source, trait, state, condition, characteristic, and property* are synonymous and were grouped together. Subsequently, the item *trait* was chosen to represent its group of items (refer to grey boxes in Figure 3). Additionally, a lexical item that has meronymous relations (part-of) with another item represents an attribute of the latter. For example, *relevance* is an item that have a meronymous relation with *project traits* (refer to round shapes in Figure 3). Lastly, some lexical item groups were considered redundant. These groups have hyponymous (kind-of) relations with other items that provide more inclusive meanings. For example, items *successful, undesirable, negative, adverse,* and *failure* are all considered as *kind-of* a variation of expected outcomes. *Success* and *failure* are considered scale extremities while *undesirable, negative, adverse* are considered antonymous points (Akmajian et al., 2001). This step in the analysis process resulted in the identification of four components of software project risk. They are risk traits, undesirable events, management practices, and expected outcomes.

Appendix H: Description of the Card Sorting Exercise – Part 2

Those methods include interviews, questionnaires, ranking-type Delphi surveys, as well as the review and analysis of existing literature. For example, the eight risk factors used in Alter and Ginzberg's (1978) study were identified using data from 60 software implementation cases that were collected via structured interviews. Based on a survey of experienced project managers, Boehm (1991) also developed a top ten risk factor list. On the other hand, Schmidt et al. (2001) followed a methodical risk factor identification process that involved iterative surveying of panels of experts from Hong Kong, Finland and the US. The process involves brainstorming and grouping of risk factors, narrowing down the identified factors into manageable lists, and rank ordering them according to their importance in regards to the success of software projects. On the other hand, Barki et al. (1993) reviewed the IS literature to identify risk factors in order to develop and test a software development project risk construct.

Risk Factor Identification Method	**References**
Delphi studies	Addison (2003), Huang et al. (2004), Keil et al. (1998), Murs et al. (2003), Pare et al. (2008), Schmidt et al. (2001).
Interviews	Alter and Ginzberg(1978), Fowler and Horan (2007), Moynihan (1996), Taylor (2006), Wallace (2004).
Surveys	Boehm (1991),Lyytinen(1988), Doherty and King(2001).
Literature review and analysis	Aloini et al. (2007), Anderson and Narasimhan (1979), Barki et al. (1993), Han and Huang(2007), Jiang et al. (2000), Kamhawi(2007).
Case Study analysis	McFarlan (1981).

Table H.1: Risk Factor Identification Methods

The list of 98 risk items was first sent to two judges. The judges were doctoral candidates in the management of information systems program at a reputable North

American university. Moore and Benbasat's (1991) recommendations were used as a guide to undertake the card sorting process. Category definitions, instructions, and an example were provided to the judges along with the pool of risk items. Afterwords, the judges' responses were analysed using SPSS. Cohen's Kappa was calculated to assess the inter-judge agreement concerning the classification. Kappa's value was 0.593 indicating a somewhat unacceptable level of agreement between the two judges. An acceptable Cohen's Kappa measure has been judged to be greater than 0.65 (Jarvenpaa 1989).

Informal meetings were conducted with the two judges in order to discuss the disagreements regarding the risk items' categorization. Their feedback helped clarify the ambiguities relating to the wording of risk items. Subsequently, the wording of some of the risk items was rectified and the list was subjected to another round of validation. The revised list was sent by e-mail to 13 project managers in the Information Technology and Services industry. Three of the five project managers had between 15 and 25 years of software project management experience. The two other judges had four and six years of project management experience respectively. Each project manager was sent an invitation message by email describing the research objective along with the revised list as an attachment. Three reminders were sent in the following two weeks to encourage participation. Eventually, five project managers responded and their feedback was analysed. Since Cohen's Kappa cannot be determined when there are more than two judges, we used Fleiss's Kappa instead. Fleiss's Kappa is a measure related to Cohen's kappa that works for any number of judges. Fleiss's kappa was 0.4521 (see table H.2) indicating a moderate level of agreement according to Landis and Koch (1977). Those results were surprising given that there was little improvement even after the list was revised and clarified.

	Agreement	Kappa	Standard Error	z-value	p-value
Overall	.7265	+.4521	.0322	+14.06	.0000
Traits Category	.7373	+.4521	.1210	+3.74	.0002
Events Category	.7149	+.4521	.1139	+3.97	.0001

Table H.2: Level of inter judge agreement – second Q-sorting round (Fleiss's Kappa)

A close look at the cases reveals that the seven judges from both the first and second round combined agree on 48 out of the 98 items. Therefore, informal meetings were conducted with two of the five judges of round two in order to discuss the disagreement regarding the rest of the items. Consequently, the feedback from those two judges, detailed comments from a third judge, and the previous feedback from the judges of the first round were analysed. Two main issues contributed to the ambiguity of the disputed items. First, the wordings of some of the items were still confusing. For example, while judges felt that some items were project risk traits in nature, those items contained words to suggest otherwise. For them, items like "project involves new or immature technology", "volatile business and organizational environment", and "staff volatility" were difficult to categorize. These items were project risk traits in nature, yet they contained words like *volatile* and *involves* which led some judges to believe that they are events. Accordingly, such items were reworded to remove such ambiguities. Second, there is a certain level of agreement between the judges that some of the items are neither project risk traits nor undesirable events since they are more related to process failures and mismanagement. For example, judges were confused when asked to categorize items like "failure to involve end-users", "failure to define clear project milestones","not managing requirement changes properly", "poor project planning", "failure to monitor project progress", "inappropriate staffing of IT personnel", or "failure to train

IT personnel on used technology". For them those items are process or management failures. So, we examined the software project risk management literature and discovered that all of the 14 items were indeed identified as risk management practices. For example, "provide training programs" is a risk management practice recommended by Alter and Ginzberg (1978) and Boehm (1991). Thus, a shortfall in providing training programs is neither a project risk trait nor an undesirable event. Instead, it is the lack of implementing a risk management practice by the project management. Categorizing those 14 items as risk factors would not have been an issue since mismanagement could be classified a risk factor. However, classifying them as traits or events does not adequately characterise them. Therefore, we removed all 14 items from our list resulting in a list of 56 project risk traits and another of 32 undesirable events (see Appendix E). The revised list was then sent to three IT project managers for another round of Q-sorting. Inter-judge agreement was reassessed using Fleiss's Kappa. A Kappa value of 0.91 was determined demonstrating an almost perfect agreement amongst the judges (see table H.3).

	Agreement	**Kappa**	**Standard Error**	**z-value**	**p-value**
Overall	.9602	+.9094	.0767	+11.86	.0000
Traits Category	.9717	+.9132	.2505	+3.65	.0003
Events Category	.9364	+.9057	.1633	+5.55	.0000

Table H.3: Level of inter judge agreement – third Q-sorting round (Fleiss's Kappa)

Appendix I: Existing Categories of Risk Management Practices

The main premise in those articles is that managers must make use of different sets of risk management strategies depending on the types of high value risks associated with their projects. In order to conceptualize the risk and risk management profiles authors favoured reviewing the related literature instead of identifying them empirically. The variables used to conceptualize risk management profiles were limited to a selected few such as planning, user participation, and coordination.

References	**Project Risk Variables**	**Project Risk Management Variables**
Andres and Zmud (2002)	Task interdependence and goal conflict.	Mechanistic and organic coordination defined according to the level of formality and cooperativeness
Barki et al. (2001)	A measure of the risk construct validated by Barki et al. (1993).	Formal planning, internal integration, and user participation.
Nidumolu (1996)	Requirement uncertainty.	Vertical and horizontal coordination.
Mathiassen et al. (2007)	Requirement identity, volatility, and complexity.	Requirement development techniques (specifying, prioritization, discovery and experimentation).

Table I.1: Risk management studies using a contingency approach

The second group includes articles that investigate the relationship between project risk management practices and project success. In general, the authors incorporated project risk as a moderating factor. Here too, project risk management variables employed in those studies were identified from the related literature and were restricted to a few such as training, prototyping, coordination, and user participation (see table I.2 for examples).

References	Project Risk Management Variables	Moderating Variables
Sharma and Yetton(2007)	Training.	Task interdependence and technical complexity.
Deephouse et al. (1996)	Planning, training, coordination, design reviews, software prototyping, cross-functional teams.	Project size, application area, and type.
McKeen(1994)	User participation.	Task complexity, system complexity, user influence, and communication

Table I.2: Risk management studies using a variance approach with risk factors as moderating variables

The third group of studies includes articles that investigated the relationship between certain risk management practices and project risk. For the most part, authors focused on risk management practices or approaches that the related literature recognized as essential to reduce project risk. Similar to the previously described groups, authors of this group selected a confined list of risk variables from the literature instead of identifying them empirically (see table I.3).

References	Project Risk Management Variables	Risk Variables
Sherif et al. (2006)	Coordination mechanisms (monitoring, communication) and organizational learning.	Goal conflict.
Ropponen and Lyytinen (2000)	Risk management methods from Boehm (1991). E.g., decisions analysis and training.	Top ten risk factor list adapted from Boehm (1991).
Zmud(1980)	Evolutionary development.	Project uncertainty and information flow.
Robey et al. (1989, 1993)	User participation.	User-IS Conflict.

Table I.3: Risk management studies using a variance approach

The last group includes studies that empirically identified risk management practices associated with software project risk factors. In order to discover risk management practices, authors used a variety of methods such as surveys, group discussion sessions, interviews, observations, and questionnaires. Hence, this stream of research provides a comprehensive and granular account of the activities that project managers perform to tackle risk. Mostly, authors of this group classified the identified mechanisms under different categories and mapped them to risk factors that were identified empirically as well. There are seven articles pertaining to this group (see table I.4).

References	**Risk Management Mechanisms Identification Method**
Boehm (1991).	Survey.
Tesch et al. (2007).	Project Management Institute group discussion sessions.
Alter and Ginzberg (1978).	Structured interviews (56 mini-case studies).
McFarlan (1981).	Observations in an unknown number of companies.
Benaroch et al. (2006).	Structured interviews, questionnaires.
Baskerville and Stage (1996), Ben-David and Raz (2001).	Case study.

Table I.4: Risk Management Mechanisms Identification Method

Appendix J: Case Survey Coding Process

This appendix describes in more detail the case survey coding process. It comprises three sections. Section 1 describes how the cases were coded. Section 2 explains the method used to convert the data into ratio variables.

Section 1

Excel was used to code the cases. A 82 x 166 table was produced. The 166 columns represent the codes we searched for. Fifty codes for project traits, 30 for undesirable events, 39 management practices [reduction], 39 management practices [mitigation], and 9 expected outcomes. After we identified text that expresses a specific code, we highlighted its position in the article and changed the value in its corresponding cell for that case from null to 1. For instance, in Davidson (2002, pg 334), the following text was identified:

> *Technical staff in the sales organization, working with system developers in ISI, IS personnel at GHI, and external consultants, conducted requirements determination activities over the course of 30 months examined here.*

This text indicates that there is a diversity of stakeholder groups which is a project risk trait. Therefore, this text was highlighted in the article and the item number is indicated (risk trait 24). Additionally, the cell in the 82x166 table that represents Davidson's case and risk trait number 24 was changed from null to x as demonstrated in the table below.

CODE ID	**65**	**24**		**23**	**31**		**37**	**22**		**13**	**22**		**13**	**31**		**29**
	Project Risk Traits				Undesirable Events			Reduction Practices			Mitigation Practices			Outcomes (Deviation)		
Case	**End users technical experience**	Diversity of stakeholder groups	**…**	**Clarity of system requirements**	**Mis-understanding of requirements**	**…**	**Scope changes**	**Share plan with users**	**…**	**Restructure project team**	**Hire external consultants**	**…**	**Restructure project team**	**Meeting quality standards**	**…**	**Degree of use**
Davidson (2002)		**x**					**x**				**x**					
Hee-Woong & Pan (2006)	**x**				**x**									**x**		

Section 2

In order to transform the coded data into ratio variables we counted the number of codes for each dimension and divided this value by the highest number of codes pertaining to the dimension found in all 82 cases. For example, in the case described in Davidson (2002), we found a total of 8 project traits. The maximum number of project traits found in any of the 82 cases we coded was 14. Therefore, the intensity of project traits in the case described in Davidson (2002) is 8/14 (0.57). We did the same for the undesirable events, expected outcomes, reduction practices, and mitigation practices and created a table of 82 rows (cases) x 5 columns (the five constructs in the causal model in figure 5).

Appendix K: Articles Included in the Case Survey

Agarwal, R., Higgins, C., and Tanniru, M. R. 1991. "Technology Diffusion in a Centralized MIS Environment: Experiences at Carrier," *Information & Management* (20:1), pp 61-70.

Ahn, J.-H., and Skudlark, A. E. 1997. "Resolving conflict of interests in the process of an information system implementation for advanced telecommunication services," *Journal of Information Technology (Routledge, Ltd.)* (12:1), pp 3-13.

Akkermans, H. 2002. "Vicious and virtuous cycles in ERP implementation: A case study of interrelations between critical success factors," *European Journal of Information Systems* (11:1), pp 35-46.

Al-Mashari, M., and Al-Mudimigh, A. 2003. "ERP implementation: Lessons from a case study," *Information Technology & People* (16:1), pp 21-33.

Anderson, D., Howell-Barber, H., Hill, J., Javed, N., Lawler, J., and Zheng, L. 2005. "a study of web services projects in the financial services industry," *Information Systems Management* (22:1) Winter2005, pp 66-76.

Apte, U., Sankar, C. S., Thakur, M., and Turner, J. E. 1990. "Reusability-Based Strategy for Development of Information Systems: Implementation Experience of a Bank," *MIS Quarterly* (14:4), pp 421-433.

Aubert, B. A., Barki, H., Patry, M., and Roy, V. 2008. "A multi-level, multi-theory perspective of information technology implementation," *Information Systems Journal* (18:1), pp 45-72.

Avison, D., and Malaurent, J. 2007. "Impact of cultural differences: A case study of ERP introduction in China," *International Journal of Information Management* (27:5), pp 368-374.

Avital, M., and Vandenbosch, B. 2000. "SAP implementation at Metalica: an organizational drama in two acts," *Journal of Information Technology (Routledge, Ltd.)* (15:3), pp 183-194.

Axtell, C. M., Waterson, P. E., and Clegg, C. W. 1997. "Problems integrating user participation into software development," *International Journal of Human-Computer Studies* (47:2), pp 323-345.

Azad, B., and Faraj, S. 2011. "Social power and information technology implementation: A contentious framing lens," *Information Systems Journal* (21:1), pp 33-61.

Bagranoff, N. A., and Brewer, P. C. 2003. "PMB investments: An enterprise system implementation," *Journal of Information Systems* (17:1), pp 85-115.

Bard, J. F., and Graham, A. 1993. "Developing a data-collection system for PCB assembly: a case study in software engineering," *Engineering Management, IEEE Transactions on* (40:2), pp 191-202.

Barker, T., and Frolick, M. N. 2003. "ERP Implementation Failure: A Case Study," *Information Systems Management* (20:4) Fall 2003, pp 43-49.

Bartis, E., and Mitev, N. 2008. "A multiple narrative approach to information systems failure: a successful system that failed," *European Journal of Information Systems* (17:2), pp 112-124.

Bhattacherjee, A. 1998. "Management of emerging technologies: Experiences and lessons learned at US West," *Information & Management* (33:5), pp 263-272.

Beynon-Davies, P., Mackay, H., and Tudhope, D. 2000. "‘It’s lots of bits of paper and ticks and post-it notes and things . . .’: a case study of a rapid application development project," *Information Systems Journal* (10), pp 195-216.

Boonstra, A., and van Offenbeek, M. 2010. "Towards consistent modes of e-health implementation: Structurational analysis of a telecare programme's limited success," *Information Systems Journal* (20:6), pp 537-561.

Bondarouk, T., and Sikkel, K. 2005. "Explaining IT Implementation Through Group Learning," *Information Resources Management Journal* (18:1), pp 42-60.

Bussen, W., and Myers, M. D. 1997. "Executive Information System Failure: a New Zealand case study," *Journal of Information Technology (Routledge, Ltd.)* (12:2), pp 145-153.

Butler, T. 2003. "An institutional perspective on developing and implementing intranet- and internet-based information systems," *Information Systems Journal* (13:3), pp 209-231.

Bygstad, B. 2003. "The implementation puzzle of CRM systems in knowledge-based organizations," *Information Resources Management Journal* (16:4), pp 33-45.

Carugati, A. 2008. "Information system development activities and inquiring systems: an integrating framework," *European Journal of Information Systems* (17:2), pp 143-155.

Chen, C. C., Law, C., and Yang, S. C. 2009. "Managing ERP Implementation Failure: A Project Management Perspective," *Engineering Management, IEEE Transactions on* (56:1), pp 157-170.

Chu, C., and Smithson, S. 2007. "E-business and organizational change: A structurational approach," *Information Systems Journal* (17:4), pp 369-389.

Chun, M., Montealegre, R., and Griffy-Brown, C. 2007. "Leveraging the Sales Force with Portal Technology at the American Subsidiary of a Japanese Motor Company Case Series (A), (B) and (C)," *Communications of the Association for Information Systems* (20:1), p 7.

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

Davidson, E., and Chiasson, M. 2005. "Contextual influences on technology use mediation: a comparative analysis of electronic medical record systems," *European Journal of Information Systems* (14:1), pp 6-18.

Devadoss, P. R., Pan, S. L., and Huang, J. C. 2003. "Structurational analysis of e-government initiatives: A case study of SCO," *Decision Support Systems* (34:3), pp 253-269.

Dong, L., Neufeld, D., and Higgins, C. 2009. "Top management support of enterprise systems implementations," *Journal of Information Technology* (24:1), pp 55-80.

Drummond, H. 1996. "The politics of risk: trials and tribulations of the Taurus project," *Journal of Information Technology* (11:4), pp 347-357.

Fedorowicz, J., Gogan, J. L., and Culnan, M. J. 2010. "Barriers to Interorganizational Information Sharing in e-Government: A Stakeholder Analysis," *Information Society* (26:5), pp 315-329.

Fitzgerald, G., and Russo, N. L. 2005. "The turnaround of the London Ambulance Service Computer-Aided Despatch system (LASCAD)," *European Journal of Information Systems* (14:3), pp 244-257.

Gosain, S., Lee, Z., and Kim, Y. 2005. "The management of cross-functional inter-dependencies in ERP implementations: emergent coordination patterns," *European Journal of Information Systems* (14:4), pp 371-387.

Gogan, J. L., and Gelinas, U. J., Jr. 2007. "Managing the Internet Payment Platform project," *Journal of Information Technology* (22:4), pp 410-419.

Heeks, R., and Stanforth, C. 2007. "Understanding e-Government project trajectories from an actor-network perspective," *European Journal of Information Systems* (16:2), pp 165-177.

Hee-Woong, K., and Pan, S. L. 2006. "Towards a Process Model of Information Systems Implementation: The Case of Customer Relationship Management (CRM)," *Database for Advances in Information Systems* (37:1), pp 59-76.

Irani, Z., Sharif, A. M., and Love, P. E. D. 2001. "Transforming failure into success through organisational learning: an analysis of a manufacturing information system," *European Journal of Information Systems* (10:1), pp 55-66.

Jensen, C. B. 2006. "The Wireless Nursing Call System: Politics of Discourse, Technology and Dependability in a Pilot Project," *Computer Supported Cooperative Work: The Journal of Collaborative Computing* (15:5/6), pp 419-441.

Joshi, K., and Lauer, T. W. 1999. "Transition and change during the implementation of a computer-based manufacturing process planning system: an analysis using the equity implementation model," *Engineering Management, IEEE Transactions on* (46:4), pp 407-416.

Keil, M. 1995. "Pulling the plug: Software project management and the problem of project escalation," *MIS Quarterly* (19:4), pp 421-447.

Kenneth, J. W., and Kozar, S. A. 1978. "Management's Role In Information System Development Failures: A Case Study," *MIS Quarterly* (2:2), pp 7-16.

Kirsch, L. J. 2004. "Deploying Common Systems Globally: The Dynamics of Control," *Information Systems Research* (15:4), pp 374-395.

Kawalek, P., and Wastell, D. G. 1999. "A case study evaluation of the use of the viable system model in information systems development," *Journal of Database Management* (10:4), pp 24-32.

Kautz, K. 2009. "Improvisation In Information Systems Development Practice," *Journal of Information Technology Case and Application Research* (11:4), pp 30-59.

Lapointe, L., and Rivard, S. 2005. "A Multilevel Model of Resistance to Information Technology Implementation," *MIS Quarterly* (29:3), pp 461-491.

Levine, H. G., and Rossmoore, D. 1993. "Diagnosing the Human Threats to Information Technology Implementation: A Missing Factor in Systems Analysis Illustrated in a Case Study," *Journal of Management Information Systems* (10), pp 55-73.

Loeb, K. A., Rai, A., Ramaprasad, A., and Sharma, S. 1998. "Design, development and implementation of a global information warehouse: a case study at IBM," *Information Systems Journal* (8:4), pp 291-311.

Löfgren, A. 2007. "Mobility In-site: Implementing Mobile Computing in a Construction Enterprise," *Communications of the Association for Information Systems* (20:1), p 3.

Luna-Reyes, L. F., Zhang, J., and Cresswell, A. M. 2005. "Information systems development as emergent socio-technical change: a practice approach," *European Journal of Information Systems* (14:1), pp 93-105.

Lyytinen, K., Newman, M., and Al-muharfi, A.-r. A. 2009. "Institutionalizing enterprise resource planning in the Saudi steel industry: A punctuated socio-technical analysis," *Journal of Information Technology* (24:4), pp 286-304.

Lui, K. M., and Chan, K. C. C. 2008. "Rescuing Troubled Software Projects by Team Transformation: A Case Study With an ERP Project," *Engineering Management, IEEE Transactions on* (55:1), pp 171-184.

Mähring, M., Keil, M., Mathiassen, L., and Pries-Heje, J. 2008. "Making IT Project De-Escalation Happen: An Exploration into Key Roles *," *Journal of the Association for Information Systems* (9:8), pp 462-496.

Mähring, M., and Keil, M. 2008. "Information Technology Project Escalation: A Process Model," *Decision Sciences* (39:2), p 239.

Maldonado, E. A., Maitland, C. F., and Tapia, A. H. 2010. "Collaborative systems development in disaster relief: The impact of multi-level governance," *Information Systems Frontiers* (12:1), pp 9-27.

Montealegre, R., and Keil, M. 2000. "De-escalating Information Technology Projects: Lessons from the Denver International Airport," *MIS Quarterly* (24:3), pp 417-447.

Nah, F. F.-H., and Delgado, S. 2006. "Critical Success Factors For Enterprise Resource Planning Implementation And Upgrade," *Journal of Computer Information Systems* (47), pp 99-113.

Newman, M., and Robey, D. 1992. "A Social Process Model of User-Analyst Relationships," *MIS Quarterly* (16:2), pp 249-266.

Nichols, M. L. 1981. "A Behavioral Analysis for Planning MIS Implementation," *MIS Quarterly* (5:1), pp 57-66.

Irani, Z., and Love, P. E. D. 2000. "The Propagation of Technology Management Taxonomies for Evaluating Investments in Information Systems," *Journal of Management Information Systems* (17:3), pp 161-177.

Newman, M., and Sabherwal, R. 1996. "Determinants of commitment to information systems development: A longitudinal investigation," *MIS Quarterly* (20:1), pp 23-54.

Niehaves, B., Klose, K., and Becker, J. 2006. "Governance Theory Perspectives on IT Consulting Projects: The Case of ERP Implementation," *e-Service Journal* (5:1), pp 5-26.

Nordheim, S. 2006. "Implementing enterprise content management: from evolution through strategy to contradictions out-of-the-box," *European Journal of Information Systems* (15:6), pp 648-663.

Pan, G., and Pan, S. L. 2006. "Examining the coalition dynamics affecting IS project abandonment decision-making," *Decision Support Systems* (42:2), pp 639-655.

Pan, G., Pan, S. L., and Newman, M. 2009. "Managing Information Technology Project Escalation and De-Escalation: An Approach-Avoidance Perspective," *Engineering Management, IEEE Transactions on* (56:1), pp 76-94.

Pan, G., and Pan, S. L. 2011. "Transition to IS Project De-Escalation: An Exploration Into Management Executives' Influence Behaviors," *Engineering Management, IEEE Transactions on* (58:1), pp 109-123.

Pardo, T. A., Cresswell, A. M., Thompson, F., and Zhang, J. 2006. "Knowledge sharing in cross-boundary information system development in the public sector," *Information Technology & Management* (7:4), pp 293-313.

Pramatari, K., Evgeniou, T., and Doukidis, G. 2009. "Implementation of collaborative e-supply-chain initiatives: an initial challenging and final success case from grocery retailing," *Journal of Information Technology* (24:3), pp 269-281.

Ray, P. K., and Ray, S. 2010. "Resource-Constrained Innovation for Emerging Economies: The Case of the Indian Telecommunications Industry," *Engineering Management, IEEE Transactions on* (57:1), pp 144-156.

Randolph, G. 2004. "Use-Cases and Personas: A Case Study in Light-Weight User Interaction Design for Amll Development Projects," *Informing Science* (7), pp 105-116.

Sillince, J. A. A., and Mouakket, S. 1997. "Varieties of political process during systems development," *Information Systems Research* (8:4), pp 368-397.

Silva, L., and Hirschheim, R. 2007. "Fighting Against Windmills: Strategic Information Systems and Organizational Deep Structures," *MIS Quarterly* (31:2), pp 327-354.

Spasser, M. A. 2002. "Realist Activity Theory for Digital Library Evaluation: Conceptual Framework and Case Study," *Computer Supported Cooperative Work: The Journal of Collaborative Computing* (11:1/2), pp 81-110.

Sutanto, J., Kankanhalli, A., Tay, J., Raman, K. S., and Tan, B. C. Y. 2008. "Change Management in Interorganizational Systems for the Public," *Journal of Management Information Systems* (25:3) Winter2008, pp 133-175.

Wastell, D. G. 1996. "The fetish of technique: methodology as a social defence," *Information Systems Journal* (6:1), pp 25-40.

Wei, H.-L., Wang, E., and Pei-Hung, J. 2005. "Understanding misalignment and cascading change of ERP implementation: a stage view of process analysis," *European Journal of Information Systems* (14:4), pp 324-334.

Appendices References

Alter, S., and Ginzberg, M. "Managing Uncertainty in MIS Implementation," *Sloan Management Review (pre-1986)* (20:1) 1978, p 23-31.

Andres, H.P., and Zmud, R.W. "A contingency approach to software project coordination," *Journal of Management Information Systems* (18:3) 2002, pp 41-70.

Barki, H., Rivard, S., and Talbot, J. "Toward an Assessment of Software Development Risk " *Journal of Management Information Systems* (10:2) 1993, pp 203-225.

Barki, H., Rivard, S., and Talbot, J. "An Integrative Contingency Model of Software Project Risk Management " *Journal of Management Information Systems* (17:4) 2001, pp 37-69.

Bélanger, F., and Carter, L. "Trust and risk in e-government adoption," *Journal of Strategic Information Systems* (17:2) 2008, pp 165-176.

Ben-David, I., and Raz, T. "An integrated approach for risk response development in project planning," *The Journal of the Operational Research Society* (52:1) 2001, p 14.

Benaroch, M., Lichtenstein, Y., and Robinson, K. "Real Options in Information Technology Risk Management: An Empirical Validation of Risk-Option Relationships," *MIS Quarterly* (30:4) 2006, pp 827-864.

Boehm, B. "Software Risk Management: Principles and Practices," *IEEE Software* (8:1) 1991, pp 32–41.

Charette, R.N. "The mechanics of managing IT risk*," *Journal of Information Technology* (11:4) 1996, pp 373-378.

Deephouse, C., Mukhopadhyay, T., Goldenson, D.R., and Kellner, M.I. "Software Processes and Project Performance," *Journal of Management Information Systems* (12:3) 1996, pp 187-205.

Drummond, H. "The politics of risk: trials and tribulations of the Taurus project," *Journal of Information Technology* (11:4) 1996, pp 347-357.

Heemstra, F.J., and Kusters, R.J. "Dealing with risk: a practical approach," *Journal of Information Technology* (11:4) 1996, pp 333-346.

Jarvenpaa, S.L. "The Effect of Task Demands and Graphical Format on Information Processing Strategies," *Management Science* (35:3), March 1, 1989 1989, pp 285-303.

Landis, J.R., and Koch, G.G. ""The measurement of observer agreement for categorical data," *Biometrics* (33) 1977, pp 159-174.

Lauer, T.W. "Software project managers' risk preferences," *Journal of Information Technology* (11:4) 1996, pp 297-295.

Liu, S., Zhang, J., Keil, M., and Chen, T. "Comparing senior executive and project manager perceptions of IT project risk: a Chinese Delphi study," *Information Systems Journal* (20:4) 2010, pp 319-355.

Mathiassen, L., Saarinen, T., Tuunanen, T., and Rossi, M. "A Contigency Model for Requirements Development," *Journal of the Association for Information Systems* (8:11) 2007, p 569.

McFarlan, F.W. "Portfolio Approach to Information Systems," *Harvard Business Review* (59:5) 1981, p 142.

McKeen, J.D., Guimaraes, T., and Wetherbe, J.C. " The Relationship Between User Participation and User Satisfaction: An Investigation of 4 Contingency Factors," *MIS Quarterly* (18:4), Dec 1994, pp 427-451.

Moore, G.C., and Benbasat, I. "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *Information Systems Research* (2:3), 1991 1991, pp 192-222.

Nidumolu, S.R. "A Comparison Of Structural Contingency And Risk-Based Perspectives On Coordination In Software Development Projects," *Journal of Management Information Systems* (13:2) 1996, pp 77-113.

Powell, P.L., and Klein, J.H. "Risk management for information systems development," *Journal of Information Technology* (11:4) 1996, pp 309-319.

Ropponen, J., and Lyytinen, K. "Components of Software Development Risk: How to Address Them? A Project Manager Survey," *IEEE Transactions on Software Engineering* (26:2) 2000, pp 98-111.

Sharma, R., and Yetton, P. "The Contingent Effects of Training, Technical Complexity, and Task Interdependence on Successful Information Systems Implementation," *MIS Quarterly* (31:2) 2007, pp 219-238.

Sherif, K., Zmud, R.W., and Browne, G.J. "Managing peer-to-peer conflicts in disruptive information technology innovations: the case of software reuse," *MIS Quarterly* (30:2) 2006, pp 339-356.

Sumner, M. "Risk factors in enterprise-wide/ERP projects," *Journal of Information Technology* (15:4) 2000, p 317.

Tesch, D., Kloppenborg, J.T., and Frolick, N.M. " IT Project Risk Factors: The Project Management Professionals Perspective," *The Journal of Computer Information Systems* (47:4) 2007, p 61.

Willcocks, L.P., Lacity, M.C., and Kern, T. "Risk mitigation in IT outsourcing strategy revisited: longitudinal case research at LISA," *Journal of Strategic Information Systems* (8:3) 1999, pp 285-314.

Zmud, R.W. "Management of Large Software Project Efforts," *MIS Quarterly* (4:2) 1980, pp 45-55.

Chapter IV – Article 3

Design Principles for software project risk management systems[13]

Mazen El-Masri
HEC Montréal
3000 Chemin-de-la-Côte-Ste-Catherine
Montréal, CANADA, H3T 2A7
Telephone number: 514 340-6000 ext. 2553
Fax Number: 514 340-6132
Mazen.el-masri@hec.ca

Suzanne Rivard
HEC Montréal
3000 Chemin-de-la-Côte-Ste-Catherine
Montréal, CANADA, H3T 2A7
Telephone number: 514 340-6493
Fax Number: 514 340-6132
Suzanne.rivard@hec.ca

[13] A preliminary version of this essay is published in the 2012 ICIS proceedings:
El-Masri M., Rivard, S., (2012) "Towards a Design Theory for Software Project Risk Management Systems", Proceedings of ICIS 2012. Orlando, Florida, December 2012.

Abstract

This paper describes the design, demonstration, and evaluation of a software project risk management system that accounts for the dynamic interactions among different components of residual risk. To identify the artifact's design principles, we refine an existing residual risk construct with the help of a panel of experts by dissecting its dimensions into subcomponents. We also conducted a case survey of 82 software projects and observed patterns of interplay among the elected residual risk components. Those patterns and the refined model were used to derive the design principles for the IT-artifact. To demonstrate the principles, we use simulation software to develop a prototype of the IT-artifact that embeds these rules in its form and function. We then assess the IT artifact's validity and utility in an intervention in the risk management of a software implementation project in an organizational setting. The IT artifact was also validated by simulating risk scenarios based on three existing IS project cases and contrasting its results with the realities described in the cases. Our results show that an IT artifact that reflects the prescribed design principles provides superior risk analysis and decision support. The paper also demonstrates a thorough design science approach that mixes rigorous methods from various disciplines to develop and evaluate design theory.

Keywords

Software project residual risk, risk management, Case survey, Design science, Simulation, Fuzzy set analysis

Introduction

Software project risk management has been acknowledged, both in research and practice, as playing a fundamental role in the success of software projects. The IS literature describes various practices that managers can adopt to assess project risk and mitigate negative impacts on project success. For instance, the analysis of key decisions was found to be an effective practice for mitigating requirement risk (Ropponen and Lyytinen 2000), and piloting, prototyping, and staging were found to be effective practices for mitigating risks such as technical complexity and project size (Benaroch et al. 2006).

Software project risk management is often described as comprising two sequential steps: risk assessment and risk control (Boehm 1991). In the first step, risk factors are identified, analyzed, and prioritized. In the second step, activities are planned and executed and project risk is monitored. However, in the past decade IS researchers have underlined the intricate and dynamic nature of software project risk suggesting that risk assessment and control activities are intertwined (e.g., Alter and Sherer 2004; Schmidt et al. 2001). For example, Alter and Sherer (2004) claimed that initial and emergent risk, management practices, and project objectives interact in unfavorable ways. Recently, empirical evidence provided support for this claim. Specifically, two studies showed that risk components pertaining to the technology, user, organizational, resource, management, etc. interact over time, causing snowball effects that divert software projects from achieving their objectives (Sicotte and Paré 2010; Warkentin et al. 2009). However, these studies are exploratory in nature. The literature does not yet recommend a theoretical framework that can explain the interactions of risk components. The software project risk constructs and other risk models that the information systems literature describes do not account for the dynamic interactions among the various components of risk.

In practice, software providers offer tools that are designed to assist managers and other involved project participants in risk assessment and planning. For the most part, these tools provide basic functionalities that support the risk management process standardized by the Project Management Institute (PMI 2004). For example, RiskyProject Professional (http://www.intaver.com/) and @RISK (http://www.palisade.com/) which lead the market in risk management software offer generic functionalities such as creating risk registers, assigning risk to tasks and resources, generating a probability-impact matrix, and performing Monte Carlo simulations. In general, these tools facilitate decision-making often by using visual representations such as Gantt charts, sensitivity analysis reports, and mathematical computations of project risk exposure. However, they do not account for interactions between risk components. For instance, the probability-impact matrix feature in @RISK treats risk factors as independent of one another to estimate project risk exposure.

With the intention to provide project risk managers with superior decision support than existing tools, this paper defines and evaluates the design principles of an IS that accounts for the risk interplay phenomenon. A design science research (DSR) approach guided our research. In the following sections, we begin by analyzing the literature that justifies our design theory development. Afterwards, we describe the DSR methodology used. Next, we explain the steps implemented to discover the design principles of the intended software project risk management system (SPRMS). Then, we describe our strategy to evaluate the design principles. In this section, we present a prototype of the IT artifact and demonstrate how it was evaluated in an organizational intervention and by simulating data of three existing IS project cases. Finally, we discuss our findings and study limitations.

Theoretical Framework

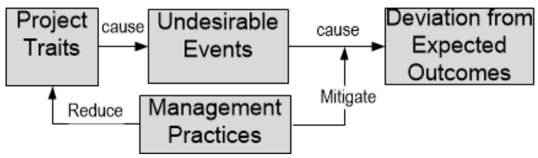
Empirical evidence suggests that the interactions of software project risk factors, undesirable events, and mismanagement over time cause dynamic snowball effects that divert software projects from achieving their objectives. For example, Sicotte and Paré (2010) who carried out a longitudinal study of two implementation projects showed how the mitigation of technological risk (technical infeasibility) by acquiring an alternative information system reasulted in augmented political risk (concerns about data privacy). The introduction of the new system also increased technical risk due to the complexity of its integration with other systems which led to delays and reduced system performance. As a result, the end-users formed negative attitudes towards the project and were dissatisfied with the system. Warkentin et al. (2009) provide other examples of how risk factors interact with one another that lead to project failure. For example, the authors show how the culture of the project organization together with flawed evaluation mechanisms caused severe delays and user resistance.

While the above evidence suggests that software project risk is dynamic, no existing software project risk construct could be used to examine this phenomenon. For the most part, existing risk constructs assume one of three conceptualizations. The first conceptualization views risk as a set of factors – properties or events – that pose a threat to the success of software projects (e.g., Jiang et al. 2007; Na et al. 2007; Wallace and Keil 2004). The second conceptualization views risk in terms of risk exposure which is the probability that an undesirable event or outcome will occur and the consequences associated with the occurrence of this event or outcome (e.g., Barki et al. 2001; Boehm and Bhuta 2008; Jia et al. 2008; Wu 2008; Wu et al. 2008). Undesirable outcomes correspond to the negative deviations from expected outcomes such as schedule and budget overruns, lower quality, and performance shortfalls (e.g.: Barki et al., 2001; Han and Huang, 2007). The last conceptualization views risk as the negative variations in project outcomes as a result of risk management's failure to respond to threats (Benaroch 2002; Benaroch et al. 2006; Clemons 1995). Here, the assessment of project risk takes into consideration the risk management practices that managers can implement in order to minimize the deviation from the expected project outcomes (Benaroch et al. 2006).

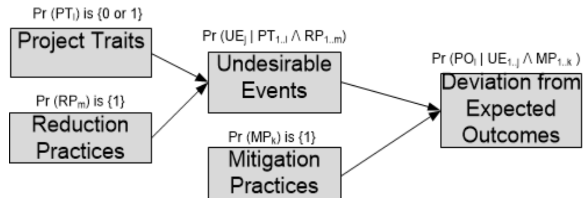
One view of risk that integrates all three conceptualizations into one construct is offered by El-Masri and Rivard (Essay 2). The authors conceptualize risk as a multidimensional construct that has four interrelated dimensions: risk traits, undesirable events, risk management practices, and expected outcomes (figure 4.1). The inclusion of risk management practices as one of the dimensions emphasizes a view of risk as residual risk. According to the authors, management practices can be reduction mechanisms applied to diminish the significance of project risk traits (e.g., training to improve staff knowledge). Alternatively, they can be mitigation mechanisms that moderate the negative impact of occurred undesirable events on project outcomes (e.g., freeze software requirements after experiencing scope creep). The authors define residual risk as the conditional probability of deviation from the expected project outcomes given that all accessible risk management practices (reduction and mitigation) are considered. They offer the following formula that links together the four dimensions in order to mathematically express the residual risk construct.

$$\text{Residual Risk (RR)}^{i} = \sum_{1..j} \text{Prob}\ [\text{PO}_{i} \mid \text{UE}_{1..j} \wedge \text{MP}_{1..k}] \times \text{Prob}\ [\text{UE}_{1..j} \mid \text{PT}_{1..l} \wedge \text{RP}_{1..m}]$$

Logical representation of residual risk



Probabilistic network representation of residual risk



Mathematical representation of residual risk

$$\text{Residual Risk (RR)}^{i} = \sum\nolimits_{1..j} \text{Prob}\ [\text{PO}_i \mid \text{UE}_{1..j} \wedge \text{MP}_{1..k}] \times \text{Prob}\ [\text{UE}_{1..j} \mid \text{PT}_{1..l} \wedge \text{RP}_{1..m}]$$

Figure 4.1 Software Project Residual Risk (El-Masri and Rivard, Essay 2)

The above software project residual risk construct can be considered comprehensive since it integrates the four main risk dimensions of the three conceptualizations identified in the IS literature. Still, it is not elaborate enough to capture the dynamic interactions that occur between components of risk at a detailed level. The further refinement of the residual risk construct's four dimensions can help in examining such interactions. Therefore, we adopt the residual risk construct of Masri and Rivard (Essay 2) as our working construct. The methods implemented to refine it are described in the following section.

Methodology

A Design science research (DSR) approach is taken to achieve the objectives of our research. DSR complements and extends IS behavioral-science research by creating and evaluating IT artifacts that reflects existing theoretical foundations. We followed the DSR methodology process model of Peffers et al. (2007). Accordingly, our research process consists of the three stages of DSR research: 1) develop, 2) demonstrate, and 3) evaluate (see figure 4.2).

In the theory development stage, we define the design principles of the intended SPRMS. First, a panel of judges helped refine our working residual risk construct. The granular specification of the construct's dimensions facilitated the investigation of the interactions between risk components. Afterwards, we distinguished the patterns in which components of residual risk interrelate by conducting a case survey of 82 IT project cases studies. To analyze these cases, we adopted a middle-ground analytical approach by imposing an existing theoretical model as our coding scheme. We interpreted the identified patterns and the refined residual risk construct and derived a set of design principles that can govern the construction of a software project risk management system.

In the theory evaluation stage, an instantiation of an IT artifact (a prototype) that conforms to the proposed design principles was developed. To construct the prototype, we used simulation software that is based on system dynamics. The system dynamics-based simulation modeling permitted us to mathematically model interlocking and temporal interrelationship between risk components. To determine the strength of relationships between the risk components required by the prototype in order to simulate risk scenarios, the data generated from the case survey was analyzed using the fuzzy-set approach.

We evaluated the utility and validity of the IT artifact in two ways. First, we intervened in the risk management of a software implementation project at a Canadian government agency. Risk analysis reports that describe the project's risk profile, possible risk scenarios, and recommended management practices were produced based on the outcomes of the software prototype. Validity was assessed in terms of the level of agreement between the risk scenarios that the prototype predicted and the ones that occurred. Artifact utility on the other hand was evaluated in terms of the percentage of recommended risk management practices that the project team decided to implement. Additionally, the IT artifact was validated by simulating possible risk scenarios based on the traits of three project cases from the IS literature and then contrasting the prototype's results with the realities that occurred.

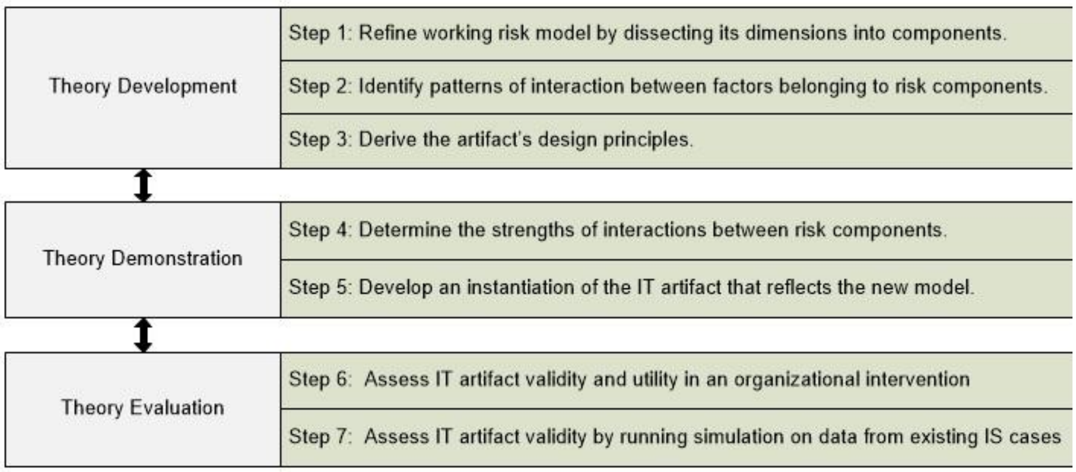


Figure 4.2 Research Methodology

Theory Development

The refinement of our working residual risk construct is a first step in order to determine the design principles of the intended IT artifact. To this end, two strategies were implemented. First, the construct's four dimensions were decomposed into subcomponents. Second, we conducted a case survey to identify patterns of interplay between risk components. The findings were then interpreted so as to deduce design principles of the intended artifact.

Refining the Software Project Residual Risk Construct

The residual risk construct we adopted defines the relationships between its four dimensions broadly. However, the extant IS literature presents numerous componentization of the software project risk construct that can be assumed. For instance, Barki et al. (1993) developed a risk construct consisting of five variables, namely technological newness, application size, team expertise, application complexity, and organizational environment. Lu and Ma (2004) conceptualized risk using policy, technical, scope, change, and management risk dimensions. Han and Huang (2007) defined risk using the user, requirement, complexity, planning and control, organizational environment and team dimensions.

While the components of the existing risk constructs slightly varied depending on the authors' theoretical perspective and research objectives, the overlaps were more evident. In order to componentize the project traits and undesirable events dimensions in our working residual risk construct, we analyzed the risk dimensions, components, and variables suggested in the IS literature and distinguished seven components of project traits and five components of undesirable events (see figure 4.3). The seven project traits components were team, user, requirements, application complexity, resources, technology, and organizational traits. As for the five undesirable events components, they were project climate, requirements, project performance, team performance and organizational related events.

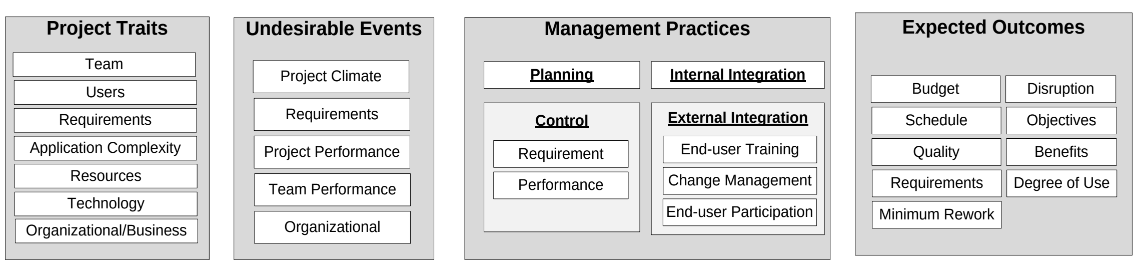


Figure 4.3 Componentization of the Software Project Residual Risk Construct Dimensions

Consequently, we asked two PhD students to classify the 58 project traits and 30 undesirable events that were attained in El-Masri and Rivard (Essay 2) study under the chosen components. This card sorting (or Q-sorting) method has been previously used in IS research (e.g., Moore and Benbasat 1991) to specify categories of constructs while maintaining convergent and discriminant validity (Straub et al. 2004). After three rounds of sorting cards, discussing classification disagreements, and clarifying the items, a satisfactory Cohen's Kappa of 0.88 (see table 4.1) was achieved. The judge's classifications are detailed in Appendix A.

Table 4.1 Inter-rater agreement of the project traits and undesirable events components

	Value	Asymp. Std. Error[a]	Approx. T[b]	Approx. Sig.
Measure of Agreement Kappa	0.875	0.037	26.105	0
N of Valid Cases	88			

The same IS literature was examined in order to specify the components of the risk management practices dimension. Studies that empirically identified risk management practices or developed and validated risk management constructs were considered. The literature describes a number of specifications of the risk management construct such as planning, control, internal integration, user participation, coordination, user training (e.g., Barki et al. 2001; Mathiassen et al. 2007; Nidumolu 1996; Ropponen and Lyytinen 2000). One eminent specification is according to four components: control, planning, internal integration, and external integration (McFarlan 1981). McFarlan's specification is widely adopted in the relevant literature (e.g., Barki et al. 2001; Deephouse et al. 1996; Liu et al. 2011a) and did not necessitate further validation. It is also comprehensive enough to organize the extant risk management practices. Accordingly, we used MacFarlan's (1981) specification of the risk management construct as a scheme to componentize the risk management practices dimension of our working residual risk construct. However, two of its four components, control and external integration, were too broad to capture risk component interactions and were refined further (see figure 4.3). Specifically, the control component was split into requirement control and project performance control. We also separated the external integration component into user participation, user training, and change management. We then used the seven components to classify the list of risk management practices identified in El-Masri and Rivard (Essay 2) (see Appendix A for the items' classification).

Finally, the nine components that form the project outcomes dimension as depicted in El-Masri and Rivard (Essay 2) were kept unchanged (see figure 4.3). This dimension has nine components: 1) adherence to budget, 2) adherence to schedule, 3) achieving the client's requirements/scope, 4) meeting quality standards, 5) amount of rework during the project, 6) degree of business disruption, 7) achieving the organization's objective, 8), yielding business benefits, and 9) degree of use. These nine components have been consistently used in the IS literature as key criteria to measure the success of the software projects. Moreover, El-Masri and Rivard (Essay 2) successfully validated a risk model using this specification of the project outcomes.

Identifying Patterns of Interactions among Residual Risk Components

The second step towards discovering the design principles of the intended IT artifact is to identify how factors belonging to the risk components interact overtime. While the risk interaction phenomenon has been recognized in recent exploratory studies (e.g., Sicotte and Paré 2010; Warkentin et al. 2009), no attempt has been made to uncover concrete patterns in which interactions occur.

One way to reveal risk interaction patterns is to scrutinize the rich case studies on software projects documented in the IS literature. To this end, we considered the case survey method to be a suitable strategy. This method is most useful when the phenomenon under study is complex and dynamic (Larsson 1993) and when a sufficient number of relevant case studies is available (Yin and Heald 1975). By applying a coding scheme, the case survey method can transform the rich data found in multiple case studies into quantitative data (Newig and Fritsch 2009). Its advantage is in its ability to generalize through the aggregation of a large number of case studies (Larsson 1993).

The cases were identified by searching the IS literature for articles that include descriptions of software project cases. ABI/INFORM Global, Business Source Complete, and IEEE Xplor were the databases searched. We also searched prominent trade publications such as IEEE Spectrum, Computerworld, and CIO Magazine. The inclusion criteria were: 1) that the study had a software project as the unit of analysis and 2) that the article provided enough information to describe the project's context and how it was conducted. We stopped analyzing new cases once we reached theoretical saturation. In total, 82 cases were analyzed (see appendix B for the complete list of articles and the search criteria used to identify the cases).

The coding scheme that was adopted represents the components of the four dimensions of the residual risk construct identified in the previous section with their associated items as the coding items (see Appendix A for the complete coding scheme). Nevertheless, we remained open to the emergence of new codes (Miles and Huberman 1994). We then coded the cases according to the described coding scheme. We searched each case description for codes that matched items in the coding scheme. Once codes were found we determined whether they were related. To detect the relationships we searched for expressions and verbs that linked two or more codes together irrespective of the dimension or component they belonged to. If two or more codes were considered related, we indicated the nature of the interrelationship and recorded the dimension and the associated component. For example, from the following text describing the case by Pan et al. (2008, p. 263), "the change of project manager has contributed to … conflicts between the new manager and some team members", we identified a relationship between two codes: change in project management and conflict within the project team. We also inferred that the relationship "contributed to" is causal. Since both codes belong to the undesirable events dimension, we recognized that there exists a causal relationship between two undesirable events. In the case of the above example, the change in project management is an undesirable event related to the organization/team structure whereas conflict within the project team is a climate related undesirable event. We added this scenario to our database. The various scenarios of interrelationships between the codes were clustered in matrices as suggested by Miles and Huberman (1994, p. 57) in order to facilitate the analysis and identification of patterns (see examples in table 4.2). Those scenarios were iteratively compared and contrasted so as to distinguish the patterns. The analysis resulted in four distinct patterns that are discussed in the subsequent section.

The vast majority of the association pattern scenarios between the residual risk dimensions that we identified typify the relationships between the dimensions of the original residual risk construct. However, our analysis also revealed four new patterns of associations between the construct's dimensions as well as among factors within the project traits and undesirable events dimensions. These patterns are: 1) the multiplicative effect of project traits, 2) the sequential effects of undesirable events, 3) the presence of a third variable, and 4) the tradeoff when selecting risk management practices (see figure 4.4).

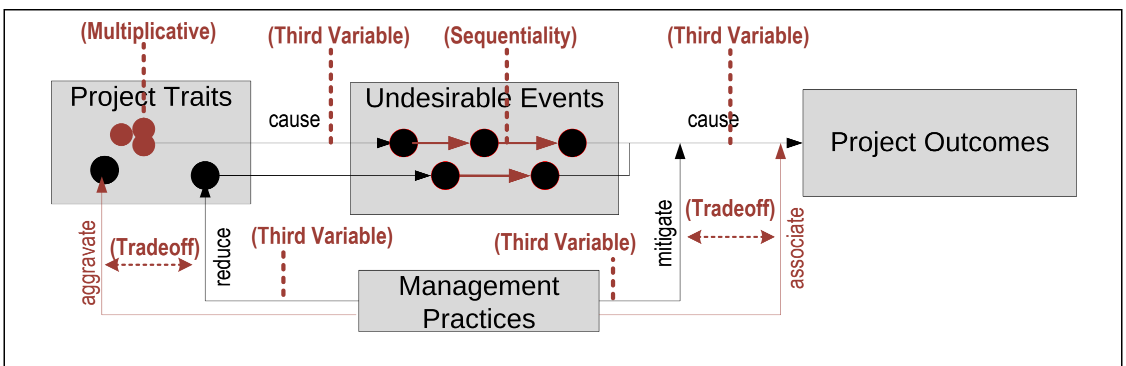


Figure 4.4 Interrelationships among the Dimensions of the Residual Risk Construct

Pattern #1: Multiplicative Effect of Project Traits

This pattern refers to situations in which the presence of more than one code pertaining to project traits significantly amplified the probability of occurrence of undesirable events. Fourteen scenarios were identified having this pattern. Our analytical process can be understood through the excerpt below from the case describing the failure of the multibillion-dollar Taurus project (Drummond 1996). Four codes (in square brackets) are identified: 1) lack of requirement clarity (project trait), 2) changes in requirements (undesirable event), 3) unanticipated rework (undesirable event), and 4) interfaces with other systems (project trait). The associations between the codes were then interpreted and displayed in matrices (see table 4.2). The following excerpt reveals how the multiplicative effect of two project traits – the lack of requirement clarity and the number of interfaces with other systems – significantly amplified the amount of unanticipated rework:

> *The critical underlying assumption, however, was that [1) the requirements were clear]. No sooner had [2) one modification been agreed in one quarter than another party demanded something]... So they (the team) were constantly saying, 'OK well we will have [3) to recode and redo] to cope with that.' Then [4) software manufacturers] would say, 'Ah but then you have [3) to make this change and you have to make that change] and you have [4) to interface these things]...'* (Drummond 1996, p. 351).

Table 4.2 Sample of the Scenarios Collected on the Multiplicative Effect of Project Traits

No	Code(s)	Dimension	Component	Association	Code(s)	Dimension	Component
1	1) Lack of requirement clarity	Traits	Requirements	Causal	2) Requirement change	Events	Requirements
2	4) Interfaces with other systems	Traits	Technology	Causal	N/A	N/A	N/A
3	2) Requirement change	Events	Requirements	Leading to	3) Unanticipated rework	Events	Team
4	2) Lack of requirement clarity and 4) interfacing with other systems	Traits	Technology and Requirements	Significantly amplified	3) Unanticipated rework	Events	Team

In other scenarios, the multiplicative effect of project traits was easily recognizable. For example, in one of the cases, the multiplicative effect of two project traits – the 1) "lack of IT staff experience" and the 2) "lack of stability of software requirements" – led to 3) slow progress (Wastell 1996):

> *They (the IS staff) argued that [3) the slow progress] was due to [1) inexperience] with the method and the CASE tool, compounded by growth in [2) the breadth and depth of requirements]* (Wastell 1996, p. 26).

The pattern of project traits having a multiplicative effect sheds some light on the intricate nature of risk. To evaluate project risk, the actors involved in risk assessment subjectively evaluate the probability of the occurrence of undesirable events. To this end, they assess a set (t_n) of project traits that could influence their occurrence (Barki et al. 1993). However, if the multiplicative effect of project traits is taken into account, this assessment could entail $2^n - 1$ permutations instead of only n permutations. In other words, a software project with only five project risk traits requires the consideration of 31 (2^5 - 1) different trait combinations which exerts a substantial cognitive load on the actors evaluating risk.

Pattern #2: Sequential Effect of Undesirable Events

This pattern represents scenarios where one undesirable event influences the occurrence of another. In total, 161 scenarios of the interrelationships we found were between undesirable events. For instance, a scenario identified in the PRISM project (Pan et al. 2008) shows that 1) management changes (an undesirable event) led to both 2) implementation problems and 3) conflicts (undesirable events).

> *[1) The change of project manager] has contributed to several [2) implementation problems]. No proper handover was made during this change of project leadership. Also, due to different styles of working,*

there were [3) conflicts] between the new manager and some team members (Pan et al. 2008, p. 263).

The risk assessment approach that is so prevalent in the IS literature does not account for the sequential effect of undesirable events. Existing approaches consider undesirable events as independent from one another. Risk exposure is often determined by multiplying the probability of occurrence of undesirable events by their associated losses (Boehm 1991). For example, Han and Huang (2007) identified the top 10 risk factors as perceived by project managers. They asked managers to assess, using a five-point Likert scale, the probability[14] of occurence of 27 risk factors (undesirable events and project traits which they classified under six risk categories) and their impacts on four project outcomes namely, cost, schedule, team, and technical performance. Consequently, the authors calculated the exposure to risk by multiplying their probability of their occurrence with their associated impacts on project outcomes (see table 4.3).

Table 4.3 Sample of the Probability, Impact, and Risk Exposure from Han and Huang (2007 p. 46)

Undesirable Event	**Probability (5 pt Likert)**	**Impact (tech)**	**Impact (cost)**	**Impact (schedule)**	**Impact (team)**	**Risk Exposure**
Requirement changes	3.13	3.12	3.30	3.27	2.67	38.68
Management changes	2.18	2.37	2.42	2.39	2.28	20.62
Conflict between stakeholders	2.35	2.27	2.40	2.44	2.06	21.55

However, Han and Huang's (2007) calculation assumes that the probabilities of occurrence of undesirable events are independent from one another. The formula they used is the following:

[14] Probability was determined using a 5-point Likert scale (1 = remote to 5 = Near certainty).

$Static\ risk\ exposure\ (UE) = \sum_{i=1}^{n} Prob\ (UE)\ x\ Impact\ (UE)_i$,where UE = undesirable event, $Impact(UE)_i$ is the impact of the undesirable event on the i_{th} project outcome.

The fact that we found a significant number of scenarios which suggest a sequential effect of undesirable events highlights the need to look at the possibility of a dynamic interplay between events as they unfold. Hence, a measure of risk exposure that reflects sequential effects should include conditional probabilities between undesirable events, which can be represented by the following equation:

Dynamic risk exposure (UE) =

$\sum_{i=1}^{n} Prob\ (UE)\ \times\ Impact\ (UE)_i + \sum_{i=1}^{n}\sum_{j=1}^{m} Prob\ (UE|UE_j)\ \times\ Impact\ (UE)_i$, where UI $\notin$ Set(UE_j).

Consequently, in order to determine probability values for a single undesirable event, actors involved in risk assessment should not only assess their project's traits, but also variations in the undesirable event's probability value given the occurrence of other events. This assessment involves (n)×(n – 1) permutations of undesirable events, and may be beyond their capacity.

Pattern #3: The Presence of a Third Variable

This pattern represents scenarios in which the relationship between two factors that contribute to residual risk is conditional on the existence of a third factor. Twelve scenarios showed a conditional relationship between factors belonging to the different risk dimensions. An example of this pattern is exhibited in a case described by Kirsch (2004):

> *When [1) communication with European counterparts became quite tense] as the project moved into development, stakeholders added [2) formal evaluation mechanisms]. These mechanisms served to encourage stakeholders to set aside their individual differences…which ultimately moved the project forward. As the Shipments project moved closer to*

> *installing the first increments, additional formal measurement and evaluation mechanisms were utilized. (Because) [3) tasks were structured and well defined], stakeholders applied the existing control mechanisms to implementation, including scheduled meetings, progress reports, project management software, and specific success criteria* (Kirsch 2004, p. 387).

Three codes were identified in the above excerpt: 1) strains in communication, 2) formal control mechanisms, and 3) well-defined tasks. The interpreted text reveals that formal control mechanisms (management practices) were implemented in order to diffuse the strains in communication arising between stakeholders (an undesirable event) and to prevent them from affecting outcomes. The success of the implemented mechanisms was conditional on the presence of a third variable (well-defined project tasks which is a project trait). Scenarios depicting this pattern were displayed in a matrix like in table 4.4.

In another scenario from Cone (2002), formal planning – a practice that has been found to positively correlate with project performance (Deephouse et al. 1996) – did not improve project performance. This was due to the fact that the time invested in formal planning was wasted due to changes in requirements. Thus, formal planning is related to project performance under the condition that requirements are stable.

> *The project was [probably too immense] and unmanageable to begin with… IBM tried to use ADA [to enforce discipline on the project] by making developers outline a design in high-level code, then fill in the blanks. But this was no match for an environment of where the FAA [kept changing its requirements]* (Cone 2002, p. 1).

This pattern highlights the complex multi-factor interplay between the various residual risk components. Actors involved in residual risk assessment not only need to evaluate residual risk as it evolves dynamically by analyzing the relationships among its components, they must also account for the presence of certain conditions in order for those relationships to occur.

Table 4.4 Sample of the Scenarios Collected Demonstrating the Presence of a Third Variable									
Case	**Code**	**Dimension**	**Category**	**Code**	**Dimension**	**Category**	**Conditional code -3rd**	**Dimension**	**Category**
Ship	Formal evaluation mechanisms	Reduction Practices	Performance control	Strains in communication	Events	Climate	Well-defined tasks	Trait	Requirements
AAS	Enforce design principles	Reduction Practices	Planning	Project size	Trait	Application Complexity	Stable requirements	Trait	Requirements

Pattern #4: The Tradeoff When Selecting Risk Management Practices

This pattern represents scenarios in which the tradeoff between positive and negative consequences is – or should be – appraised prior to the implementation of risk management practices. The failure to balance both types of consequences may result in ill-informed decisions and lead to adverse outcomes. Thirteen such scenarios were found. Those scenarios demonstrate that, even though the implementation of risk management practices reduced the significance of certain project traits or mitigated the impact of undesirable events, they also increased the implication of other project traits or triggered new undesirable events. One of the recurring scenarios involved: 1) end-users' involvement; a practice that project managers implement 2) to clarify requirements and 3) ensure end-users' acceptance of the system. When involving end-users, the project managers evaluated – or failed to evaluate – the tradeoff between the probability that end users would reject the system and the probability that 4) new requirements would emerge as a result of their involvement. An excerpt from a case described by Sillince and Mouakket (1997) demonstrates this tradeoff scenario:

> *After [1) meeting the secretary-administrators several times] and getting more [2) knowledge of the system and the flow of information], the two analysts realized that the system could not be developed without taking account of their requirements… That meant [4) additional new requirements]… The analysts were not prepared for this change… but acknowledged the fact that the system needed to be expanded to include some requirements of the end-users if they wanted the system to be practical and to be [3) accepted by the end-users]* (Sillince and Mouakket 1997, p. 376).

Another scenario in a case described by Cone (2002) shows how managers failed to assess the tradeoff between system quality and system acceptance due to end-user involvement.

> *...by giving them the opportunity to customize the [display colors], the design introduced the possibility that they might accidentally make planes appear invisible against the background* (Cone 2002, p. 2).

This pattern underscores the value of the different practices available to managers. Benaroch et al. (2006) assert that the value of a project's residual risk exposure should be offset by the options available to managers. However, due consideration must be given to both the positive and negative impacts of those options.

Design Principles for Software Project Risk Management Systems

The four patterns of interrelationships among residual risk components show that software project residual risk is multifaceted and dynamic. This requires that the actors involved in residual risk assessment take into account the interactions and conditions of interactions between its components; forecast how future possible events trigger one another and amplify residual risk; and evaluate the negative and positive impacts of available mitigation practices. This problem-solving activity may be beyond a project risk manager's capacity, resulting in miscalculations and lack of foresight. For example, if the actors involved in risk assessment have to flawlessly evaluate project residual risk exposure based on ten project traits and their multiplicative effects then 1023 ($2^{10} - 1$) distinct combinations of project traits should be considered. This issue can be addressed with a risk management system design that considers those interactions during project residual risk assessment. Accordingly, we propose the design science paradigm as an alternative approach. We believe that project risk managers can be better supported using an IT artifact tailored to the specificities and intricacy of software projects.

Design science is a problem-solving paradigm used to provide practical solutions to real-world problems, and design theories are prescriptive in nature (Hevner et al. 2004; Walls et al. 1992). However, design theories must be justified by normative and descriptive theories – known as kernel theories – and integrated "into design paths intended to produce more effective information systems" (Walls et al. 1992). The first step towards building our design theory is to translate our findings on the four patterns of risk factor interrelationships into design principles. To stay faithful to the precepts of design science research, we were guided by the guidelines of Gregor and Jones (2007) on design theory construction. Accordingly, the overarching purpose of our research is to provide a practical solution that the actors involved in residual risk assessment can use to evaluate software project risk. Project failures are often attributed to management's inability to forecast temporal interactions between risk factors and accurately assess project risk exposure (Charette 2005). Hence our design principles must address those interactions so that the IT artifact can make a more accurate forecast of project residual risk exposure. We propose four preliminary design principles in keeping with the identified risk factor interaction patterns.

The Association Principle

Upon determining the probability of occurrence of undesirable events, the IT artifact should allow for possible associations between project traits, as well as between undesirable events. As mentioned above, the existing literature acknowledges four types of associations (El-Masri and Rivard 2010). First, a project trait is associated with one or more undesirable events. This association is required to determine the probability of occurrence of undesirable events. For example, the lack of clarity of software requirement (a project trait) can be lead to events such as requirement changes and scope creep. Second, an undesirable event is associated with one or more project outcomes. This association is necessary to identify anticipated adverse impacts on project outcomes. For example, requirement changes can result in schedule and budget overruns. Third, management practices are associated with project traits. These associations are required to determine the decrease in the probability of occurrence of undesirable events. For instance, additional efforts to analyze client and end-user needs can clarify software requirements thereby reducing the probability or requirement changes and scope creep. Fourth, management practices are associated with undesirable events. These associations are necessary to determine the reduction in the impacts of undesirable events on project outcomes. For instance, scaling down the software requirements reduces the adverse impact of requirement changes on the project's budget and schedule. The risk factor association principle prescribes two novel ways in which risk factors ought to be associated and that the IT artifact should allow for:

1) Associations between project traits: When two or more project traits are simultaneously associated with at least one undesirable event. This is required in order to reassess the probability of occurrence of an undesirable event, given that the project possesses certain configurations of project traits. For instance, one of the scenarios identified in the case survey shows that the lack of requirement clarity together with the anticipated level of interfacing with other systems significantly amplified the amount of unexpected rework required (Drummond 1996, p. 351).
2) Associations between undesirable events: When undesirable events are associated with other undesirable events. This type of association is required in order to reassess the probability of occurrence of undesirable events, given that the probability of occurrence of other associated undesirable events is greater than zero. For example, a scenario identified in the case survey shows that changes in the project's management structure led to conflicts between stakeholders, as well as to implementation problems (Pan et al. 2008, p. 263).

The Regulation Principle

The IT artifact should be able to regulate the interplay between factors belonging to the different risk dimensions. There are two types of regulations: conditional and tradeoff. Conditional regulations refer to the IT artifact's ability to regulate associations between factors by attributing conditions to their interactions. In other words, the association between two factors is conditional on a third factor. For instance, a scenario from the case described by Cone (2002, p. 1) shows that formal planning can mitigate risk due to project size given the condition that the software requirements are stable. On the other hand, "tradeoff regulations" refers to the artifact's ability to regulate the anticipated gain generated by planned management practices by balancing their positive and negative impacts on project traits or project outcomes. For example, the case described by Sillince and Mouakket (1997, p. 376) shows that, involving the various end-user groups requires the analysis of the tradeoff between their level of satisfaction and the permitted increase in the number of software requirements.

The Simulation Principle

The IT artifact should be able to simulate possible project risk scenarios in order to: (1) evaluate the project's residual risk exposure, (2) prioritize project traits according to their impact on project residual risk exposure, and (3) suggest which management practices can help reduce the project's residual risk exposure. Prior to performing simulation analysis, the project manager should select and rank the project's expected outcomes. For instance, the project manager could specify that respecting the budget, satisfying end-user needs, and producing high quality software are the most important outcomes respectively. Afterwards, the actor involved in residual risk assessment selects, from a list of traits, those that are deemed risky and rate their significance (e.g.: project size, staff expertise). Residual risk assessment is achieved by running a simulation that uses the rated project traits, as well as the ranked expected outcomes as inputs. To this end, the IT artifact makes use of a database of past projects. This database should contain previously identified associations between the factors belonging to different residual risk components. The results from the case survey we conducted can serve as the initial data source. Once the simulation has been performed, the project manager can choose the suitable management practices proposed by the artifact, evaluate their cost and benefits, and rerun the simulation. The simulation can be rerun with different combinations of management practices until an optimal solution is attained.

The Self-Learning Principle

The IT artifact should have the ability to learn from past risk management experiences. As the simulation principle commends, the artifact must make use of a database of past projects in order to simulate possible risk scenarios, measure the probability of their occurrence and their impact on project outcomes. With new information being added to this database, the IT artifact must automatically adjust. To this end, it must be able to compute probability and impact values of risk scenarios and risk management practices at runtime each time a simulation is performed.

Theory Demonstration

Design science research (DSR) produces IT artifacts that are expressed using various levels of abstractions. Constructs, methods, and models in DSR are concrete IT artifacts that can generally be converted into instantiations. IS design theories are viewed as abstract IT artifacts and are described with design principles (Gregor and Hevner 2013).

In DSR, design principles can evolve from experimenting with prototypes that feature assumptions on the models, methods and constructs embedded in the intended IT artifacts (e.g., Markus et al. 2002). Once the authors advanced the prototypes into a useful solution, the design principles were interpreted from the embedded assumptions. Conversely, IT artifacts can be advanced deductively from existing social and natural science research. For example, McLaren et al. (2011) specified the initial prototype of a measurement model of multilevel strategic fit using existing models in social science.

Regardless of the approach taken – inductive or deductive, a form of artifact instantiation is required to evaluate the DSR knowledge. Instantiations are problem-specific aggregates of constructs, models, and/or methods (Winter 2008). They can be questionnaires (e.g., McLaren et al. 2011) or complete software prototypes (e.g., Markus et al. 2002) so long as their utility and validity can be evaluated (Gregor and Hevner 2013).

Because the intended IS must implement complex formulas to determine residual risk probabilities, we favored to demonstrate the IT artifact using simulation software. The IT artifact instantiation served as a "proof-of-concept" that helped to evaluate the design principles. However, the strengths of the interrelationships between the different residual risk components are an essential component of the intended IT artifact. The next two sections describe the method used to quantify the residual risk component interrelationships and the resulting instantiation of the IT artifact.

Determining the Strength of Interrelationships between Components of Residual Risk

In the information systems literature, software project risk is commonly viewed as a notion of the future or, more specifically, a problem that could occur and negatively impact the project's objectives (Boehm 1991; Charette et al. 1997; Drummond 1996). However, one of the primary difficulties of investigating risk lies in the difficulty in forecasting the occurrence of problematic events. This compels researchers to ask project stakeholders to subjectively estimate the probability of occurrence of undesirable events (e.g., Han and Huang 2007) or use project traits as proxy measures in order to estimate event probabilities (e.g., Barki et al. 1993). Nevertheless, limiting the forecasting estimates to the project stakeholders' perceptions and experiences overlooks all other possible risk scenarios that are not yet unknown to them. According to Kahneman (2011), there is a tendency to believe that human-made decisions and choices are based on reasoning and mental computations. However, it is often intuitions and feelings that are the main sources of the decision making process.

One approach that was recently suggested is to adjust project predictions based on experiences from other similar projects in order to reduce errors in forecasting and probability calculations (Kahneman 2011). According to Kahneman (2011), accurate predictions can be best achieved by combining the project's "inside view" represented by its traits with the "outside view" which is an external reference to experiences of other projects that manifest similar traits. The author endorses implementing the "outside view" using a forecasting method proposed by Flyvbjerg et al. (2005) in which a large database that provides information on statistics of different components of past projects is created. This database can then be used to adjust predictions based on the particularities of the project under consideration.

The data collected from the case survey of 82 software project cases can be used as the database that provides statistics on the strengths of the interrelationships between the different risk components. Therefore, we statistically analyzed this data using the fuzzy set qualitative comparative analysis technique (Ragin 2009). This Boolean based Set-Theoretic Analysis technique is particularly appropriate to validate complex models (Fiss 2011). Its premise is that, in complex models, causal relations are better understood in terms of relationships between sets rather than correlations (Ragin 2009). Cases are conceptualized as combinations of attributes where these very combinations give the cases their uniqueness (Fiss 2011). This technique has gained recent popularity in social science research (e.g., Fiss 2011; Rivard and

Lapointe 2012) since it bridges the divide between quantitative and qualitative research methods.

Each of the 82 software project cases was considered a specific configuration of an IS project representing a possible combination of causal conditions and effects. As described earlier, conducting the case survey allowed for the detection of patterns in which risk components of the four residual risk dimensions interact among each other. Additionally, it provided the number of identified codes for every component in each of the 82 cases. A table of 82 rows and 35 columns was created that contains the number of items that were found for each component. The 35 columns represent the total number of components of all four dimensions combined. As previously stated, there were seven project traits, five undesirable events, seven risk reduction practices, seven risk mitigation practices, and nine project outcomes components. However, consistent with the adopted specification of the residual risk construct, we differentiated the seven risk management practice components into risk reduction and risk mitigation practices components in order to capture the practices that were preformed to reduce the significance of risk traits and the ones that were performed to mitigate the impact of undesirable events after they occurred. For instance, in the case described by Mangan and Kelly (2009), we identified one application complexity risk trait and one organizational risk trait (see figure 4.5). The case did not describe any risk reduction practice that could reduce the significance of those two traits. Additionally, two climate related events, one requirement related event and two project performance events were identified. Lastly, one mitigation practice [15] pertaining to performance control was also identified. Pertaining to the project outcomes described in the Mangan and Kelly (2009) case, it was a total failure and the project did not achieve any of its expected outcomes. Accordingly, this configuration of the Mangan and Kelly (2009) case is illustrated in figure 4.5 below.

[15] In the case survey, all practices that were implemented after the occurrence of undesirable events to mitigate their impact were considered mitigation practices.

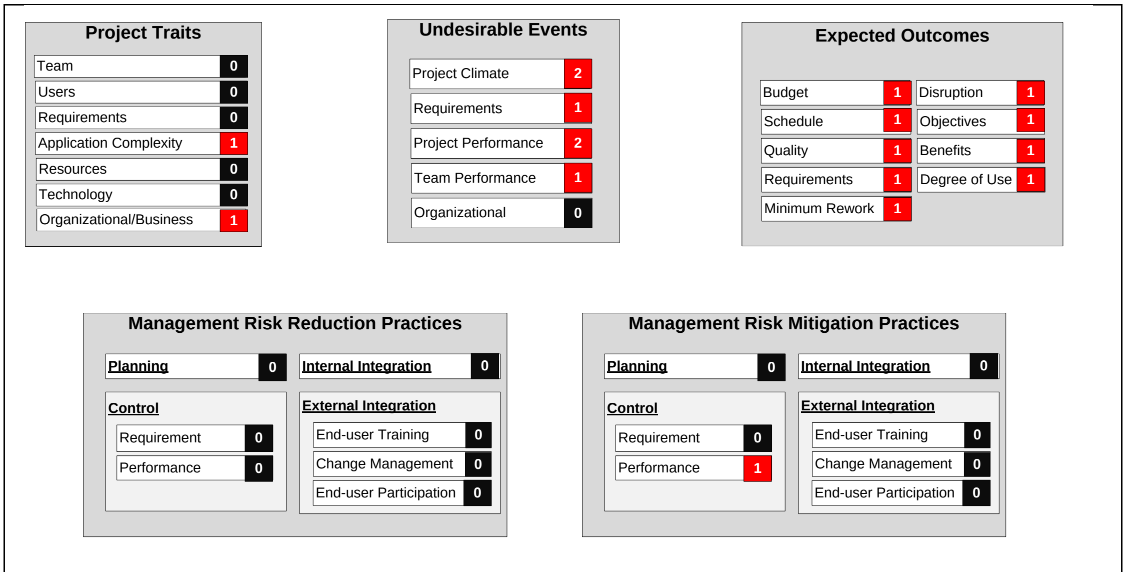


Figure 4.5 Mangan and Kelly (2009) Case Configuration

Once the coding was completed and the 82x35 table was created, we measured the degree of membership of every IS project in each of the 35 risk components (fuzzy sets in QCA vocabulary).

A fuzzy set scales the degree of membership in an interval from 0.0 to 1.0, with 0.0 indicating full exclusion from a set and 1.0 indicating full inclusion (Ragin 2009). Thus, we calibrated each risk component by counting the number of codes of each risk component found in an IS project case. We then divided that count by the largest count of codes of that risk component found in the 82 cases. For instance, if the highest number of codes in the 82 cases that belong to the 6-item requirement traits component (or fuzzy set) was 3, then all the cases (or projects) that had 3 codes identified were considered to have full membership (1.0), whereas the ones that had 1 or 2 codes identified were considered to have partial membership (0.3 and 0.7 respectively). Cases that had no identified requirement traits codes had no membership (0.0) in the requirement traits risk component. For instance, there was only one item pertaining to the application complexity component in Mangan and Kelly (2009) while the maximum number of application complexity items identified in all 82 cases was 4. Therefore, the application complexity component in Mangan and Kelly (2009) case was calibrated to 0.25 (see figure 4.6).

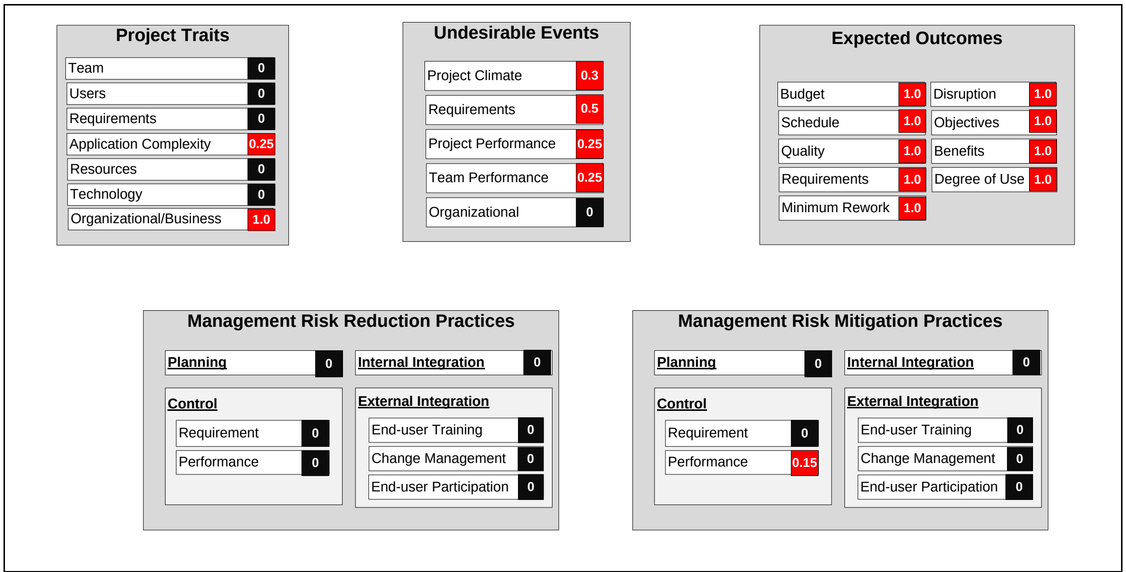


Figure 4.6 Degree of Membership of Risk Components of Mangan and Kelly (2009) Case

The 82 x 35 fuzzy-set truth table was then analyzed using the fs/QCA software in order to calculate the strengths of the relationships between the risk components. Using fs/QCA we can determine the degree of membership of outcome variables based on a configuration of calibrated causal conditions. If we take the case described in Mangan and Kelly (2009) as an example, the causal conditions for each undesirable event component (the first level of outcome variables) are the seven project traits and seven reduction practices components. Likewise, the causal conditions for each project outcome component (the second level of outcome variables) are the five undesirable events and seven mitigation practices components. The analysis we conducted using fs/QCA provided us with the degree of membership of each of the undesirable event components given the full membership of different configurations of project risk traits and risk reduction practices components. We interpreted the degree of membership of each of the undesirable events component as the probability of its occurrence given the presence of a set of project traits and the implementation of a set of risk reduction practices. Further analysis also provided the degree of membership of each of the expected outcome components given the full membership of different configurations of undesirable and risk mitigation practices components. An example of the results obtained from conducting fuzzy set analysis can be found in table 4.5 below.

Table 4.5 Sample of the Results Obtained from Fuzzy Set Analysis

Project Traits Components			**Reduction Practices**	**Undesirable Events**			
User	Technology	Complexity	End-user training	Requirement	Climate	Team Performance	Project Performance
1	1	1	1	0.71	0.67	0.67	0.53

In this example, we interpreted that the full presence of user, technology, and complexity traits components (all other traits components are absent) given that only end-user training was performed results in 71%, 67%, 67%, and 53% probabilities that requirement, climate, team performance, and project performance related events respectively will occur. Similarly, we interpreted the degree of membership of each of the project outcomes as their probability of deviation. These values were the coefficients of the mathematical equations that define the risk components' interrelationships that we embedded in the simulation model. Each equation corresponds to a specific configuration of relationships between the 35 risk components. A total of 487 different equations prescribing the strengths of relationships between risk components and conditions on their relationships were produced by the fuzzy-set analysis.

Producing an Instantiation of the IT Artifact

Due to the dynamic nature of the interactions among the components of the risk construct, simulation software that can model dynamic systems was required. We chose iThink by ISEE systems. The iThink simulation software can mathematically model the behavior of complex systems by allowing interlocking, and temporal relationships between its components. It has been previously used to study dynamic models in social science (e.g., Black et al. 2004; Clark et al. 2007). Using iThink, we were able to develop the refined residual risk construct and specify the relationships among its components. Figure 4.7 shows a subset of the complete model.

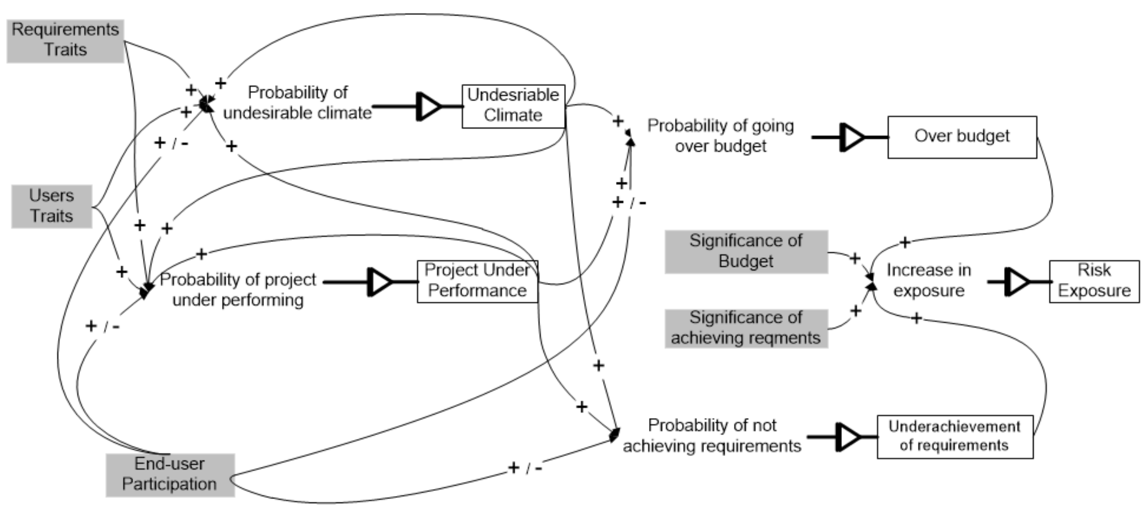


Figure 4.7 A Subset of the Software Project Risk Management iThink Simulation Model

In order to determine residual risk, we used the formula provided by El-Masri and Rivard (Essay 2): $\text{Residual Risk (RR)}^{i} = \sum_{1..j} \text{Prob}\,[\text{PO}_i \mid \text{UE}_{1..j} \wedge \text{MP}_{1..k}] \times \text{Prob}\,[\text{UE}_{1..j} \mid \text{PT}_{1..l} \wedge \text{RP}_{1..m}]$. The $\text{Prob}\,[\text{PO}_i \mid \text{UE}_{1..j} \wedge \text{MP}_{1..k}]$ denotes the probability (i.e., the degree of membership) of project outcome I given the occurrence (i.e., full membership) of a configuration of undesirable events (1..j) as well as the implementation (i.e., full membership) of a configuration of mitigation practices (1..k). $\text{Prob}\,[\text{UE}_j \mid \text{PT}_{1..l} \wedge \text{RP}_{1..m}]$ denotes the probability (i.e., the degree of membership) of undesirable event j given the occurrence (i.e., full membership) of a configuration of project traits (1..l) as well as the implementation (i.e., full membership) of a configuration of reduction practices (1..m).

Theory Evaluation

Notwithstanding the implications of considering different abstractions of IT artifacts to IS research, *theory* remains the most essential knowledge contribution to be evaluated. To this end, the utility and validity of the physical IT artifact that reflects the theory's design principles must be demonstrated (Hevner et al. 2004). To establish artifact utility is to show that the goals that the IT artifact achieved have value (March and Smith 1995). Conversely, validity in design science research denotes that the IT artifact does what it intends to do (Gregor and Hevner 2013).

An intervention in the risk management of a software implementation project in a real organizational setting was conducted in order to determine the IT artifact's utility and validity. We also evaluated the IT artifact's validity by assessing the accuracy of the IT artifact's predictions in comparison with three recent[16] IS project cases chosen randomly from the IS literature.

Intervention in the Management of Risk of a Software Implementation Project

A software implementation project (ULTIMA) at Alpha was followed for a period of 10 months. Alpha is a Canadian government agency comprised of six engineering divisions. ULTIMA is an Enterprise Content Management System that Alpha intends to use in order to consolidate the diverse types of engineering documents and standardize the document management processes. The agency

[16] The IS literature between 2011 and 2012 was searched using the same search criteria that was used for the original case survey.

planned to pilot the IS in one of the engineering departments within a six months time frame (June to December 2012) and roll it out to all other divisions by the end of 2013.

We conducted four meetings with the project team. Each meeting lasted between 2 and 3 hours. During the first two meetings, the six project team members collaboratively described the project traits and expected outcomes. A questionnaire was used to record the team's input (see Appendix C). To evaluate project risk, we used the information that the team provided as inputs for the IT artifact and ran the simulation assuming that no risk management practices were planned. We also performed a number of simulation scenarios for which we instructed the program to account for different combinations of risk management practices. We assessed all combinations in order to arrive at the optimal set of practices that were anticipated to best reduce the project's residual risk exposure which was calculated using the formula provided in El-Masri and Rivard (Essay 2). Based on the program's outputs, a risk assessment report was compiled and sent to the project manager. The report primarily contained information on the project's risk profile, the probabilities of occurrence of undesirable events and outcomes, as well the suggested management practices to mitigate risk (see Appendix D for a detailed risk report).

In the third and fourth meeting we assessed the *utility* and *validity* of the IT artifact. During these meetings, we asked the project team to describe the undesirable events that occurred and the risk management practices that were implemented or planned. We assessed validity by comparing the undesirable events with those predicted by the IT artifact. Utility was appraised in terms of the number of risk management practices that the project team planned/implemented from the list of practices that the IT artifact suggested.

The IT artifact predicted that the project had a medium level of residual risk exposure (28.3/82.5) and was more likely to succeed than to fail. It also provided the probabilities of occurrence of different types of undesirable events (see table 4.6).

Moreover, it predicted that respecting the project's schedule was the least likely outcome to happen (61%). Lastly, it considered that user involvement, project performance control, and requirements control were the most optimal management strategies to reduce risk (see table 4.7 for the complete list).

The largest number of events that occurred was related to the climate, team performance, and project performance (see table 4.6). The events that were most echoed by team members were (1) tense relationships between project stakeholders and conflict with the supplier (climate), as well as (2) technical difficulties and performance issues (team performance). The project encountered a number of delays and the deadline to complete the pilot was postponed twice; once until March and another time until June 2013 (project performance). These undesirable events are consistent with the types of events predicted which signifies the validity of the IT artifact.

To appraise the utility of the IT artifact, we contrasted the practices that were implemented (or planned) with the three types of practices suggested by the IT artifact. The project team implemented around half of the suggested risk management practices (7 out of 15) related to user involvement, project performance control, and requirement performance control (see table 4.7). Additionally, three other suggested risk management practices were planned to be implemented in the future. The fact that the project manager implemented a large percentage of the practices that were suggested was interpreted as indicating the IT artifact's utility. Indeed, after the implementation of the risk management practices, fewer undesirable events occurred. For instance, to resolve the conflict with the suppliers and reduce the team's frustrations and resolve technical problems with the IS, the project manager escalated the issues to higher management. After negotiating with the supplier, the supplier provided bug fixes that resolved major technical problems. The supplier also promised prompt response to future problems.

Additionally, the project manager and project champion perceived that the report provided by the IT artifact was useful when asked to judge its value. They found that the report provided effective insights about the likelihood of occurrence of undesirable events. Similarly, they found that the practices suggested were useful and their implementation helped control project risk. According to the project champion:

> *The risk assessment report was insightful and we used it to control the project risk. My intuition was to implement most of the risk management practices that were suggested even before I saw the report. What the report did is confirm that my intuition was correct which gave us confidence to apply the practices suggested.*

Table 4.6 A comparison between predicted and occurred events at time t = 1 (December 11, 2012) and time t = 2 (March 19, 2013)

Event type	**Estimated probability of occurrence**	**Possible events**	**Occurred t=1 (yes/no)**	**Echoed most by team at t=1**	**Occurred at t=2 (yes/no)**	**Echoed most by team at t=2**
Climate-related	100%	User resistance				
		Tension in relationships	☒	☒	☒	☒
		Frustration and team moral	☒			
		Conflict with stakeholders (supplier)	☒	☒		
		Failure to obtain stakeholder's buyin				
Team performance related	61%	Inadequate design				
		Incorrect functionality or configuration	☒			
		Technical difficulties	☒	☒		☒
		Performance related problems	☒	☒		
		Shortfall in efficiency				
		Unanticipated rework of components				
		Gold-plating				
		Failure to integrate with other IS				
		Failure to migrate from legacy IS				
Requirement	66%	Continuous changes to requirements			☒	☒

-related		Requirements misunderstanding				
		Scope creep	☒			
		Scope mis-estimation	☒			
		Inadequate business process				
Project performance related	77%	Changes in project time table	☒	☒		☒
		Shortfall in funding				
		Escalating project cost				
		Escalating project time of completion	☒			☒
Organizational/team related events	31%	Loss of top management support				
		Changes in org. management				
		Changes in organizational priorities				
		Resignation of key project member				
		Reallocation of project members				
		Changes in project management			☒	

Table 4.7. A comparison between suggested and implemented risk management practices at time t1 = 12/11/12 and t2= 03/19/13

Risk Management Practices	**Implemented time t = 1**	**Planned time t = 1**	**Implemented time t = 1**	**Planned time t = 2**
User involvement:				
Promote the software to users by providing regular information on the planning and progress of the project.				
Allow users to use the software before the launch date and take into account their concerns and ideas.		☒		☒
Define the acceptance criteria with the users (representative).	☒			
Develop and revise operational scenarios with the users.		☒		
Validate the functionality and business processes with the user representative	☒			
Project performance control:				
Conduct regular assessment of the progress of the project.	☒		☒	
Establish benchmarks and milestones.	☒		☒	
Select a champion for the project.	☒			
Involve senior management in major decisions.			☒	
Project requirements control:				
Prioritize user requirements.				
Reduce the scope of user requirements, if necessary.	☒			
Avoid changes to requirements established by the users.				
Minimize software customization.				
Map the computer requirements to business processes.		☒	☒	
Conduct non-functional requirements testing.	☒		☒	

Assessing Artifact Validity

A second strategy was implemented to validate the IT artifact. Three cases were randomly selected from the existing IS literature. The cases were coded using the same approach that was implemented in the case survey. We used the data we coded on the project's risk traits and the implemented risk management practices for each of the three projects as inputs in the IT artifact and ran the simulation. Consequently, we compared the artifact's predictions with the undesirable events and the ensuing outcomes. Specifically, for each of the three projects, we contrasted the probabilities of occurrence of each type of undesirable events with the number of undesirable events that transpired. We also contrasted the probability of not achieving the intended outcomes with the actual projects' outcomes.

As shown in table 4.8, the predictions of our IT artifact are consistent with the actual outcomes in all three IS project cases. The higher the probabilities of occurrence of undesirable events that the artifact predicted were, the more undesirable events occurred and vice versa. For instance, in the Balint (2011) case, the largest number of undesirable events that occurred were related to the climate and project performance (4 and 4 events respectively). This is consistent with the high probabilities of occurrence of these types of undesirable events that our IT artifact predicted (85% and 53% respectively).

The probability of not attaining the expected outcomes predicted by the IT artifact also seemed plausible. For the most part, the cases exhibited project performances that are coherent with those the IT artifact predicted. For instance, the E-government implementation case by Chan et al. (2011) explains that one of the only failures encountered was the acute disruptions to existing user operations once the software was in place. This is consistent with the artifact's predictions.

Moreover, the implementation of risk reduction practices reduced the probabilities of undesirable events. In the Teo et al. (2011) case, management implemented a number of risk reduction management practices. No undesirable events occurred during the project. Therefore, when the implemented practices were included in the simulation, our IT artifact predicted very low probabilities of occurrence of all types of undesirable events (4- 8%).

Table 4.8 Results of the Analysis of the three cases

Case	Risk Profile Low= 1.0 → High= 5.0							Implemented Risk Management Practices (Yes=✓, No=✗)						
	User	Team	Technology	Requirements	Resources	Complexity	Organizational	Internal integration	Planning	Requirements control	Performance control	Change management	User training	User participation
A	1.0	1.0	1.8	1.0	1.0	1.0	1.0	✓	✓	✓	✗	✓	✓	✓
B	1.3	1.0	3.8	1.0	1.0	3.6	1.0	✗	✓	✗	✗	✓	✗	✓
C	1.2	1.0	2.9	1.4	1.0	3.1	2.4	✗	✗	✗	✗	✗	✗	✗

Case	Undesirable Events Predicted (Probability in %)					Undesirable Events Occurred (# of events)				
	Climate related	Team performance related	Requirements related	Project performance	Org./team structure	Climate related	Team performance related	Requirements related	Project Performance	Org./Team structure
A	7	6	8	8	4	0	0	0	0	0
B	35	20	27	33	24	2	2	0	1	0
C	85	53	58	61	37	4	4	3	3	2

Case	Outcomes Predicted (Probability in %)								Outcomes Occurred (Yes=✓, No=✗, N/A=?)							
	Budget overrun	Schedule overrun	Lack of use	Not achieving objectives	Not realizing benefits	Not fulfilling requirements	Operational disruption	Achieving quality standards	Budget overrun	Schedule overrun	Lack of use	Not achieving objectives	Not realizing benefits	Not fulfilling requirements	Operational disruption	Achieving quality standards
A	15	24	12	15	7	15	9	5	?	?	✗	✗	✗	✗	?	✗
B	23	32	16	23	15	19	26	13	?	?	✗	✗	✗	?	✓	?
C	59	50	31	51	43	34	53	41	✓	✓	✓	✓	✓	✓	✓	✓

Case A: Teo et al. (2011) Case B: Chan et al. (2011) Case C: Balint (2011)

Discussion

This research responds to calls to examine the dynamic interactions of risk components previously voiced in the IS literature (Alter and Sherer 2004; Schmidt et al. 2001). Recent studies provided support for such interactions (Sicotte and Paré 2010; Warkentin et al. 2009). Those studies enriched our understanding of how different components of risk interacted overtime. More importantly, the studies' findings emphasize two important points. First, the overall reliance on the examination of project risk factors independently of one another in order to determine project risk and its effect on project outcomes provides only a partial indication of how the project will unfold. The interaction effects among risk components play a critical role in deciding the eventual outcomes. Moreover, the direction a software project takes is highly contingent on how undesirable events snowball overtime as well as the actions taken by management to remedy detrimental situations and steer the project back to its intended course (Sicotte and Paré 2010). The second point is related to the difficulty to distinguish key themes in which specific risk components interact in order to plan appropriately. A closer look at the scenarios of interactions among risk components that were recognized in both Sicotte and Paré (2010) and Warkentin et al. (2009) studies show that those themes are dynamic and contextual. For instance, Warkentin et al. (2009) found that organizational (including political) risk increases technical risk, whereas Sicotte and Paré (2010) found that technical risk increases political risk.

This research took a different approach to investigate the interactions of risk components. We sought an understanding of the structure of interactions between different components of residual risk and provided a mechanism (an IT artifact) that can help examine residual risk holistically. The recognition of patterns of interactions between components of residual risk irrespective of their type (user, requirements, team, etc.) instead of concrete themes of interactions that are type specific helped us to theoretically define a structure of the residual risk construct that allows for the examination of the interrelationships. The identified patterns helped in the development of a set of rules that a software project risk management system must conform to in order to account for residual risk component interactions. We believe that an IT artifact that respects these rules will provide superior decision support to project managers than existing risk management tools or methodologies that lack the necessary features to address the residual risk component interaction phenomenon. For instance, the significance of the regulation principle was evident during the intervention at Alpha. When we evaluated the impact of different combinations of risk management practices components at Alpha, our simulation revealed that the implementation of all possible types of risk management practices would not be beneficial. In certain scenarios, formal planning such as detailed project plans or cost and schedule estimation increased the overall project risk exposure. Equally, internal integration practices such as personnel training or hiring experienced consultants did not always decrease the overall exposure to risk.

The premise behind the simulation principle is that software project residual risk is intricate and multifaceted. There are numerous components that interact when examined over time. The existing knowledge on software project risk is not sufficient to suggest particular themes of interactions between risk components and their corresponding effect on project outcomes. Hence, it seems more suitable to examine risk components as configurations and allow historical data of software projects decide the strengths of interactions among residual risk components in order to provide predictions and decision support. Certainly, this approach requires that historical project data be available. The data obtained from the case survey of 82 software projects can be used as an initial database for practitioners.

It is worth mentioning that the case survey data we analyzed can indeed provide themes of interaction between risk components. For instance, our data suggests that when team performance and requirement related events occur, the probability of project performance related events transpiring increases by 16% and 12% respectively. However, this interaction is not reciprocal. Project performance related events, if occurred, increase the probability that requirement related events would occur by 3% and were found to not have an effect on team performance. Moreover, organizational related events (resignation, management changes, loss of top management support, etc.) when they occur have a strong recursive relationship (10% increase in probability).

Limitations

This study has a number of limitations. First, we used secondary data to identify patterns of interrelationships among the components of the software project residual risk construct. The authors' descriptions of the cases that we analyzed might not have completely portrayed all risk scenarios that occurred during the software projects. We minimized the significance of this limitation by only choosing the cases that provided sufficient information to describe the software project.

The second limitation relates to the inability to assess the weight of project risk traits in the cases we coded. Predominantly, the case descriptions did not indicate the significance of the project traits and we were only able to detect whether a trait was present or not. Yet, in the intervention we conducted, we assessed project traits according to their importance and we used these assessments as inputs into the simulation program. Accordingly, the significance of project trait components ranged between 0 and 5. This limitation was addressed in two ways. First, the fuzzy set analysis we conducted produced formulas where the degrees of membership of each of the undesirable event component as functions of the full membership of different configurations of project risk trait components (and risk reduction practices components). Therefore, we assumed that full membership of a given project trait component corresponds to scenarios where the trait component is most significant (5/5). So we adjusted the formulas produced by the fuzzy set analysis that we embedded in the simulation program by incorporating the inputted significance of each project trait component as its coefficient.

The previous limitation is not limited to project traits. For the most part, the cases surveyed did not indicate the degree of impact of risk management practices or the extent of deviation from expected outcomes. However, we feel that the fact that our database aggregates 82 cases has minimized the inaccuracies of interrelationships between the various components.

Moreover, our measure of artifact utility is subjective. We assumed that the amount of risk management practices that the project manager adopted based on the IT artifact's recommendations implied that the latter was useful. While this is not necessarily the case in all situations, we believe that this limitation does not have an effect on our findings. The manager of the project we intervened in had over 20 years of experience. Moreover, the project champion was also involved in deciding on the management practices to implement. S/he, like the project manager, considered the artifact's recommendations useful. Finally, all project team members were present during the meetings to discuss the artifact's recommendations. We saw no signs of discontent or disapproval.

Another limitation relates to the degree of completeness of our database. Not all configurations of project traits and risk reduction practices were present in our database since this would require 2^{14} (7 project traits and 7 risk reduction practices components) or 16384 different configurations. This number of configurations was infeasible to attain. To address this limitation we instructed the simulation program to choose the closest set of configurations, calculate the probability of occurrence of undesirable events components based on each configuration independently, and then average the resulting probabilities.

Lastly, in order to evaluate the learning principle, it was necessary that the IT artifact merge data on new projects into its existing knowledge base. This was not feasible in our study since it requires that we successively follow a number of software projects and update the IT artifact's knowledge base after every project. However, since our overarching principle is that the statistical and ordered exploitation of past project experiences is a superior mechanism to evaluation and manage risk, it seems logical that new experiences should alter future evaluations.

Conclusion

The evaluation and management of software project risk, while taking into account how risk components interact over the course of the project, requires substantial analytical efforts that managers might find cognitively overwhelming. The IS literature is marked with failure stories of large software project that were managed by very experienced managers. According to the 2009 CHAOS report by Standish group only 2% of IS projects over $10 million succeed. These statistics signal the inability to manage complex projects using static methods. Techniques like risk factor lists to conduct risk analysis do not allow for accurate evaluations of risk interactions. Such a task requires advanced algorithms that can exploit a sizeable knowledge base.

There is no extant theoretical framework that accounts for these interactions and existing project risk management software lacks functionalities that consider their impacts. This leads to risk being misrepresented and inadequately managed. Our research aims to assist management in complex analytical and decision-making processes. Specifically, it draws attention to the dynamic interplay between software project risk components and offers an alternative solution (an IT artifact) that addresses such complexities.

In terms of research, our study examines the core subject matter, the IT artifact, in the IS discipline (Weber 2003). It proposes and evaluates the design of future software that can be beneficial to the field of IT. According to Benbasat and Zmud (1999), information systems research should focus on topics that are relevant to IT field by engaging with practitioners in order to thoroughly understand their needs. Research that is stimulated by practical problems has more tendency to produce implementable implications (Benbasat and Zmud 1999). Our research is consistent with this stance. It addresses the software project risk management problem which is considered important to practice. By adopting a design science approach, the separation between research and practical implications is significantly narrowed. First, our research advances theories in information systems design by proposing design principles of future software. Design theories are native IS theories and that the IS field promotes (Straub 2012). We also produced implementable and practical implications by demonstrating and evaluating those principles. Taking an action research approach allowed us to work closely with practitioners in order to validate the proposed design principles and assess the artifact's usefulness in practice.

Furthermore, this study is in line with the recent call to address complex organizational problems using an intradisciplinarity approach. In his most recent editorial comments, the Editor-in-Chief of MIS Quarterly Paulo Goes (2013) stated that the future goals of the journal will promote intradisciplinarity research that brings together design, behavioral, and organizational sciences in order to "develop, test, and validate the desirable proof-of-concept, and the higher levels of proof-of-value and proof-of-use of technology-based systems". Lacking the required knowledge to achieve our research objectives, our study combined research from social, decision, and design sciences in order to develop and provide evidence for the proof-of-value and proof-of-use of the design of a software project risk management system. To do so, we combined research methods rooted in social science (case survey), design science (simulation and action research), and mathematics (set theoretic).

Our study adopted the design science research process suggested by Peffers et al. (2007) that aims at developing, demonstrating, and evaluating IT artifacts. By intervening in the risk management of a software implementation project, we were able to evaluate the IT artifact and measure the usefulness of its outputs. Other organizations can use the prototype or develop their own software based on the associated design principles specified in this research. Organizations can adopt the refined multifaceted risk model to exploit their own past project experiences in order to assess risk more accurately and create actionable decision knowledge.

Future research can take two directions. First, we consider this study as an initial step towards a practical software project risk management system. Nevertheless, a simulation program was used to assess our design principles instead of an initial version of the software. Hence, the proposed software features were not directly evaluated in practice. Instead, reports that were based on the simulation program's outputs were the features assessed. Future research can construct a software representation that reflects the proposed features and evaluate other aspects such as its ease of use or address other components of the design theory such as the artifact's mutability or additional design principles. Second, we believe that the understanding of software project risk that resulted from this study can be generalized and applied in other project risk phenomena. The higher-level four-dimensional risk model can be applied to other contexts such as finance, marketing, or engineering. The model's interrelated dimensions (traits, events, outcomes, and management practices) could be refined differently according to the associated field of study. Moreover, the proposed design principles can be tested and developed further in other contexts.

References

Abdel-Hamid, K. T., Sengupta, K., and Swett, C. 1999. "The impact of goals on software project management: An experimental investigation," *MIS Quarterly* (23:4), pp 531-555.

Agarwal, N., and Rathod, U. 2006. "Defining 'success' for software projects: An exploratory revelation," *International Journal of Project Management* (24:4), pp 358-370.

Akmajian, A., R, D., Farmer, A., and Harnish, R. 2001. *Linguistics: An Introduction to Language and Communication*, (5th ed.) MIT Press: Cambridge, MA.

Aladwani, M. A. 2002a. "An integrated performance model of information systems projects," *Journal of Management Information Systems* (19:1), pp 185-210.

Aladwani, M. A. 2002b. "IT project uncertainty, planning and success: An empirical investigation from Kuwait," *Information Technology & People* (15:3), pp 210-226.

Alberto Espinosa, J., Cummings, J. N., and Pickering, C. 2011. "Time Separation, Coordination, and Performance in Technical Teams," *Engineering Management, IEEE Transactions on* (PP:99), pp 1-13.

Alioni, D., Dulmin, R., and Mininno, V. 2007. "Risk management in ERP project introduction: A Review of the Literature," *Information & Management* (44:1), pp 547–567.

Alter, S., and Ginzberg, M. 1978. "Managing Uncertainty in MIS Implementation," *Sloan Management Review (pre-1986)* (20:1), p 23.

Alter, S., and Sherer, S., A 2004. "A General, But Readily Adaptable Model of Information System Risk," *Communications of the Association for Information Systems* (14:1), pp 1-28.

Ambler, S. 2007. "IT Project Success Rates Survey," http://www.ambysoft.com/surveys/success2007.html.

Anderson, J., and Narasimhan, R. 1979. "Assessing Project Implementation Risk: A Methodological Approach," *Management Science* (25:6) June 1, 1979, pp 512-521.

Andres, H. P., and Zmud, R. W. 2002. "A contingency approach to software project coordination," *Journal of Management Information Systems* (18:3), pp 41-70.

Armstrong, C., and Sambamurthy, V. 1999. "Information Technology Assimilation in Firms: The Influence of Senior Leadership and IT Infrastructures," *Information Systems Research* (10:4), pp 304-327.

Atsu, M. Y., Andoh-Baidoo, F. K., Osatuyi, B., and Amoako-Gyampah, K. 2010. "An Exploratory Study of The Contextual Factors That Influence Success of ICT Projects in Developing Nations: A Case Study of A Telecommunications Company in Ghana," *Journal of Information Technology Case and Application Research* (12:3), pp 56-81.

Aubert, B., Barki, H., Michel, P., and Roy, V. 2008. "A Multi-level, Multi-theory Perspective of Information Technology Implementation," *Information Systems Journal* (18), pp 45-72.

Baccarini, D. 1999. "The logical framework method for defining project success," *Project Management Journal* (30:4), pp 25-32.

Balint, B. 2011. "Difficulties in Enterprise System Implementation: The Case of Millicent Homesbfdgfd," *Journal of Information Technology Case and Application Research* (13:3), pp 72-84.

Banker, R. D., Bardhan, I., and Ozer, A. 2006. "Understanding the Impact of Collaboration Software on Product Design and Development," *Information Systems Research* (17:4), pp 352-373.

Barki, H. 2008. "Thar's Gold in Them Thar Constructs," *Database for Advances in Information Systems* (39:3), pp 9-20.

Barki, H., and Hartwick, J. 2001. "Interpersonal Conflict and its Management in Information System Development," *MIS Quarterly* (25:2), pp 195-228.

Barki, H., Rivard, S., and Talbot, J. 1993. "Toward an Assessment of Software Development Risk " *Journal of Management Information Systems* (10:2), pp 203-225.

Barki, H., Rivard, S., and Talbot, J. 2001. "An Integrative Contingency Model of Software Project Risk Management " *Journal of Management Information Systems* (17:4), pp 37-69.

Bartis, E., and Mitev, N. 2008. "A multiple narrative approach to information systems failure: a successful system that failed," *European Journal of Information Systems* (17:2), pp 112-124.

Basten, D., Joosten, D., and Mellis, W. 2011. "Managers' Perceptions of Information System Project Success," *The Journal of Computer Information Systems* (52:2), pp 12-21.

Bélanger, F., and Carter, L. 2008. "Trust and risk in e-government adoption," *Journal of Strategic Information Systems* (17:2), pp 165-176.

Ben-David, I., and Raz, T. 2001. "An integrated approach for risk response development in project planning," *The Journal of the Operational Research Society* (52:1), p 14.

Benaroch, M. 2002. "Managing Information Technology Investment Risk: A Real Options Perspective," *Journal of Management Information Systems* (19:2), pp 43-84.

Benaroch, M., Lichtenstein, Y., and Robinson, K. 2006. "Real Options in Information Technology Risk Management: An Empirical Validation of Risk-Option Relationships," *MIS Quarterly* (30:4), pp 827-864.

Black, L. J., Carlile, P. R., and Repenning, N. P. 2004. "A Dynamic Theory of Expertise and Occupational Boundaries in New Technology Implementation: Building on Barley's Study of CT Scanning," *Administrative Science Quarterly* (49:4), pp 572-607.

Boehm, B. 1991. "Software Risk Management: Principles and Practices," *IEEE Software* (8:1), pp 32–41.

Boehm, B., and Bhuta, J. 2008. "Balancing Opportunities and Risks in Component-Based Software Development," *IEEE Software* (25:6), pp 56-63.

Bryant, C. D., and Peck, D. L. 2007. *21st century sociology: a reference handbook*, (1st ed.) SAGE Publications.

Burton-Jones, A., and Straub, D. W., Jr. 2006. "Reconceptualizing System Usage: An Approach and Empirical Test," *Information Systems Research* (17:3), pp 228-246.

Chan, C. M. L., Hackney, R., Pan, S. L., and Chou, T.-c. 2011. "Managing e-Government system implementation: a resource enactment perspective," *European Journal of Information Systems* (20:5), pp 529-541.

Chang, K.-c., Sheu, T. S., Klein, G., and Jiang, J. J. 2010. "User commitment and collaboration: Motivational antecedents and project performance," *Information and Software Technology* (52:6), p 672.

Charette, R. N. 1989. *Software Engineering Risk Analysis and Management*, (McGraw-Hill.

Charette, R. N. 1996. "The mechanics of managing IT risk*," *Journal of Information Technology* (11:4), pp 373-378.

Charette, R. N. 2005. "Why Software Fails," *IEEE Spectrum* (42:9), pp 42–49.

Chen, C. C., Liu, J. Y.-C., and Chen, H.-G. 2011. "Discriminative effect of user influence and user responsibility on information system development processes and project management," *Information and Software Technology* (53:2), p 149.

Clark, T. D., Jr., Jones, M. C., and Armstrong, C. P. 2007. "The Dynamic Structure of Management Support Systems: Theory Development, Research Focus, and Direction," *MIS Quarterly* (31:3), p 579.

Clemons, E. K. 1995. "Using Scenario Analysis to Manage the Strategic Risks of Reengineering," *Sloan Management Review* (36:4), pp 61-71.

Cockburn, A. 2002. *Agile software development*, (1st ed.) Addison Wesley Professional.

Cone, E. 2002. "The Ugly History of Tool Development at the FAA," in *Baseline Magazine*, http://www.baselinemag.com/article2/0,1540,794112,00.asp.

Cooke-Davis, T. 2002. "The real success factors on projects," *International Journal of Project Management* (20:3), pp 185-190.

Cruse, D. A. 2004. *Meaning in Language: An Introduction to Semantics and Pragmatics*, (2nd ed.) Oxford University Press.

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

Davis, R. C. 2002. "Calculated risk: A framework for evaluating product development," *MIT Sloan Management Review* (43:4), p 71.

de Bakker, K., Boonstra, A., and Wortmann, H. 2012. "Risk managements' communicative effects influencing IT project success," *International Journal of Project Management* (30:4), p 444.

Deephouse, C., Mukhopadhyay, T., Goldenson, D. R., and Kellner, M. I. 1996. "Software Processes and Project Performance," *Journal of Management Information Systems* (12:3), pp 187-205.

DeLone, H. W., and McLean, R. E. 2003. "The DeLone and McLean model of information systems success: a ten-year update," *Journal of Management Information Systems* (19:4), pp 9-30.

Deutsch, M. S. 1991. "An exploratory analysis relating the software project management process to project success," *Engineering Management, IEEE Transactions on* (38:4), pp 365-375.

Doherty, N. F., and King, M. 2001. "An investigation of the factors affecting the successful treatment of organisational issues in systems development projects," *European Journal of Information Systems* (10:3), p 147.

Drummond, H. 1996. "The politics of risk: trials and tribulations of the Taurus project," *Journal of Information Technology* (11:4), pp 347-357.

Dvir, D., Raz, T., and Shenhar, J. A. 2003. "An empirical analysis of the relationship between project planning and project success," *International Journal of Project Management* (21:2), pp 89-95.

Edwards, J. R. 2001. "Multidimensional constructs in organizational behavior research: An integrative analytical framework," *Organizational Research Methods* (4:2), pp 144-192.

El-Masri, M., and Rivard, S. 2010. "Specifying the Software Project Risk Construct," in *Proceedings of the Sixteenth Americas Conference on Information Systems*: Lima, Peru.

El-Masri, M., and Rivard, S. Essay 2. "Specifying the Software Project Risk Construct," in *Proceedings of the Sixteenth Americas Conference on Information Systems*: Lima, Peru.

Espinosa, J. A., DeLone, W., and Lee, G. 2006. "Global boundaries, task processes and IS project success: a field study," *Information Technology & People* (19:4), pp 345-370.

Faraj, S., and Sambamurthy, V. 2006. "Leadership of Information Systems Development Projects," *IEEE Transactions on Engineering Management* (53:2), pp 238-249.

Fiss, P. C. 2011. "Building Better Causal Theories: A Fuzzy Set Approach to Typologies in Organization Research," *Academy of Management Journal* (54:2), p 393.

Flyvbjerg, B., Skamris Holm, M. K., and Buhl, S. L. 2005. "How (in) accurate are demand forecasts in public works projects?: The case of transportation," *Journal of the American Planning Association* (71:2), pp 131-146.

Fowler, J. J., and Horan, P. 2007. "Are Information Systems' Success and Failure Factors Related? An Exploratory Study," *Journal of Organizational and End User Computing* (19:2), p 1.

Freeman, E. R. 2010. "Managing for Stakeholders: Trade-offs or Value Creation," *Journal of Business Ethics* (96), pp 7-9.

Freeman, E. R., Rusconi, G., Signori, S., and Strudler, A. 2012. "Stakeholder Theory(ies): Ethical Ideas and Managerial Action," *Journal of Business Ethics* (109:1), pp 1-2.

Freeman, R. E. 2004. "The Stakeholder Approach Revisited," *Zeitschrift fuer Wirtschafts- und Unternehmensethik* (5:3), pp 228-241.

Gardiner, D. P., and Stewart, K. 2000. "Revisiting the golden triangle of cost, time and quality: The role of NPV in project control, success and failure," *International Journal of Project Management* (18:4), pp 251-256.

Gartner 2010. "IT Metrics: IT Spending and Staffing Report."

Gefen, D., Rigdon, E. E., and Straub, D. W. 2011. "Editor’s Comment: An Update and Extension to SEM Guidelines for Administrative andSocial Science Research," *MIS Quarterly* (35:2), pp iii-xiv.

Goes, P. 2013. "Editor's comments," *MIS Quarterly* (37:1), pp iii-vii.

Gopal, A., and Gosain, S. 2010. "The Role of Organizational Controls and Boundary Spanning in Software Development Outsourcing: Implications for Project Performance," *Information Systems Research* (21:4), pp 960-982.

Gregor, S., and Hevner, A. R. 2013. "Positioning and Presenting Design Science Research for Maximum Impact," *MIS Quarterly* (37:2), pp 337-355.

Gregor, S., and Jones, D. 2007. "The Anatomy of a Design Theory.," *Journal of the Association for Information Systems* (8:5), pp 312–335.

Hahn, I., Bredillett, C., Kim, G.-M., and Taloc, M. 2012. "Agility of Project Manager in Global IS Project," *The Journal of Computer Information Systems* (53:2), pp 31-38.

Hahn, J. H., Mentz, E., and Meyer, L. 2009. "Assessment Strategies for Pair Programming," *Journal of Information Technology Education* (8), pp 273-284.

Hakkinen, L., and Hilmola, O. P. 2008. "ERP evaluation during the shakedown phase: lessons from an after-sales division," *Information Systems Journal* (18:1) Jan, pp 73-100.

Han, W. M., and Huang, S. J. 2007. "An Empirical Analysis of Risk Components and Performance on Software Projects.," *Journal of Systems and Software* (18:1), pp 42-50.

Heemstra, F. J., and Kusters, R. J. 1996. "Dealing with risk: a practical approach," *Journal of Information Technology* (11:4), pp 333-346.

Henderson, J. C., and Lee, S. 1992. "Managing I/S Design Teams: A Control Theories Perspective," *Management Science* (38:6), pp 757-777.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design science in Information Systems Research.," *MIS Quarterly* (28:1), pp 75–105.

Hsu, J. S.-C., Lin, T.-C., Zheng, G.-T., and Hung, Y.-W. 2012. "Users as knowledge co-producers in the information system development project," *International Journal of Project Management* (30:1), p 27.

Kahneman, Daniel. 2011 *Thinking, fast and slow*. Macmillan.

Jackendoff, R. S. 1990. *Semantic Structures*, (2nd ed.) Mit Press.

Jarvenpaa, S. L. 1989. "The Effect of Task Demands and Graphical Format on Information Processing Strategies," *Management Science* (35:3) March 1, 1989, pp 285-303.

Jia, Z., Lihua, G., and Jia, H. Year. "Application of Support Vector Machine Based on Rough Sets to Project Risk Assessment (RS-SVM)." International Conference on Computer Science and Software Engineering2008, pp. 508-511.

Jiang, J. J., and Klein, G. 1999. "Risks to different aspects of system success," *Information & Management* (36:5), pp 263-272.

Jiang, J. J., Klein, G., Beck, P., and Wang, T. G. E. 2007. "Lack of Skill Risks to Organizational Technology Learning and Software Project Performance," *Information Resources Management Journal* (20:3), pp 32-45.

Jiang, J. J., Klein, G., and Chen, H.-G. 2001. "THe relative influence of IS project implementation policies and project leadership on eventual outcomes," *Project Management Journal* (32:3), pp 49-55.

Jiang, J. J., Klein, G., and Ellis, T. S. 2002. "A measure of software development risk," *Project Management Journal* (33:3), pp 30-41.

Julien, P. A., and Ramangalahy, C. 2003. "Competitive Strategy and Performance of Exporting SMEs: An Empirical Investigation of the Impact of Their Export Information Search and Competencies," *Entrepreneurship Theory and Practice* (27:3), pp 227-245.

Kamhawi, M. E. 2007. "Critical Factors for Implementation Success of ERP Systems: An Empirical Investigation from Bahrain," *International Journal of Enterprise Information Systems* (3:2), pp 34-49.

Kearns, S. G. 2007. "How the Internal Environment impacts Information Systems Project Success: An Investigation of Exploitative and Explorative Firms," *The Journal of Computer Information Systems* (48:1), pp 63-75.

Keil, M., Cule, P., Lyytinen, K., and Schmidt, R. 1998. "A Framework for Identifying Software Project Risks," *Communications of the ACM* (41:11), pp 76-83.

Kirsch, L. J. 2004. "Deploying Common Systems Globally: The Dynamics of Control," *Information Systems Research* (15:4), pp 374-395.

Landis, J. R., and Koch, G. G. 1977. ""The measurement of observer agreement for categorical data," *Biometrics* (33), pp 159-174.

Larsson, R. 1993. "Case survey methodology: Quantitative analysis of patterns across case studies," *Academy of Management Journal* (36:6), p 1515.

Lauer, T. W. 1996. "Software project managers' risk preferences," *Journal of Information Technology* (11:4), pp 297-295.

Law, C. H., and Ngai, E. W. 2007. "ERP systems adoption: An exploratory study of the organizational factors and impacts of ERP success," *Information & Management* (44:4), pp 418-432.

Law, K. S., Wong, C.-S., and Mobley, W. H. 1998. "Toward a taxonomy of multidimensional constructs," *Academy of Management* (23:4), pp 741-755.

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), p 87.

Lee, K.-H., Yang, G., and Graham, J. L. 2006. "Tension and trust in international business negotiation: American executives negotiating with Chinese executives," *Journal of International Business Studies* (37:5), pp 623–641.

Liang, H., Saraf, N., Hu, Q., and Xue, Y. 2007. "Assimilation of Enterprise Systems: The Effect of Institutional Pressures and the Mediating Role of Top Management," *MIS Quarterly* (31:1), pp 59-87.

Liberatore, M. J., and Luo, W. 2010. "Coordination in Consultant-Assisted IS Projects: An Agency Theory Perspective," *IEEE Transactions on Engineering Management* (57:2), p 255.

Liberatore, M. J., and Pollack-Johnson, B. 2003. "Factors influencing the usage and selection of project management software," *Engineering Management, IEEE Transactions on* (50:2), pp 164-174.

Linberg, R. K. 1999. "Software developer perceptions about software project failure: A case study," *The Journal of Systems and Software* (49:2), pp 177-192.

Liu, j., Wang, Q., and Ma, Q. 2011a. "The effects of project uncertainty and risk management on IS development project performance: A vendor perspective," *International Journal of Project Management* (29:7), p 923.

Liu, J. Y.-C., Chen, H.-G., Chen, C. C., and Sheu, T. S. 2011b. "Relationships among interpersonal conflict, requirements uncertainty, and software project performance," *International Journal of Project Management* (29:5), p 547.

Liu, S., Zhang, J., Keil, M., and Chen, T. 2010. "Comparing senior executive and project manager perceptions of IT project risk: a Chinese Delphi study," *Information Systems Journal* (20:4), pp 319-355.

Lu, X. N., and Ma, Q. G. Year. "Risk analysis in software development project with owners and contractors," Engineering Management Conference, 2004. Proceedings. 2004 IEEE International2004, pp. 789-793 Vol.782.

Lucas, W. A. 1974. "The case survey method: Aggregating case experience," Rand Corporation., Santa Monica, CA.

Lyytinen, K. 1988. "Expectation Failure Concept and Systems Analysts' View of Information System Failures: Results of an Exploratory Study," *Information & Management* (14:1), p 45.

Lyytinen, K., Mathiassen, L., and Ropponen, J. 1998. "Attention Shaping and Software Risk: a Categorical Analysis of Four Classical Risk Management Approaches," *Information Systems Research* (9:3), pp 233–255.

MacKenzie, S. B., Podsakoff, P. M., and Podsakoff, N. P. 2011. "Construct Measurement and Validation Procedures in MIS and Behavioral Research: Integrating New and Existing Techniques," *MIS Quarterly* (35:2), p 293.

Mahaney, R. C., and Lederer, A. L. 2011. "An Agency Theory Explanation of Project Success," *The Journal of Computer Information Systems* (51:4), pp 102-113.

Mangan, A., and Kelly, S. 2009. "Information systems and the allure of organisational integration: a cautionary tale from the Irish financial services sector," *European Journal of Information Systems* (18:1), p 66.

March, S. T., and Smith, G. F. 1995. "Design and Natural Science Research on Information Technology," *Decision Support Systems* (15:4), pp 251-266.

Markus, M. L., and Tanis, C. 2000. "The Enterprise System Experience - From Adoption to Success," in *Framing the Domains of IT Management: Projecting the Future Through the Past*, Pinnaflex Educational Resources inc.: Cincinnati, Ohio, United States, pp. 173-207.

Markus, M. L., Axline, S., Petrie, D., and Tanis, C. 2000. "Learning from adopters' experiences with ERP: problems encountered and success achieved," *Journal of Information Technology* (15), pp 245-265.

Markus, M. L., Majchrzak, A., and Gasser, L. 2002. "A design theory for systems that support emergent knowledge processes," *MIS Quarterly* (26:3), p 179.

Martinez Sanchez, A., and Perez, M. P. 2004. "Early warning signals for R&D Projects: An empirical study," *Project Management Journal* (35:1), pp 11-23.

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), p 569.

McFarlan, F. W. 1981. "Portfolio Approach to Information Systems," *Harvard Business Review* (59:5), p 142.

McKeen, J. D., Guimaraes, T., and Wetherbe, J. C. 1994. " The Relationship Between User Participation and User Satisfaction: An Investigation of 4 Contingency Factors," *MIS Quarterly* (18:4) Dec, pp 427-451.

McLaren, T. S., Head, M. M., Yuan, Y., and Chan, Y. E. 2011. "A Multilevel Model for Measuring Fit Between A Firm's Competitive Strategies and Information Systems Capabilities," *MIS Quarterly* (35:4), p 909.

Miles, M. B., and Huberman, A. M. 1994. *Qualitative Data Analysis: An Expanded Sourcebook*, (2nd ed.) Sage Publications: Newbury Park, CA.

Moore, G. C., and Benbasat, I. 1991. "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *nformation Systems Research* (2:3) 1991, pp 192-222.

Na, K., Simpson, J. T., Li, X., Singh, T., and Kim, K. 2007. "Software Development Risk and Project Performance Measurement: Evidence in Korea," *The Journal of Systems and Software* (80:4), pp 596-605.

Naveed, S. 1996. "An empirical test of the contingency approach to user participation in information systems development," *Journal of Management Information Systems* (13:1), pp 145-166.

Nelson, R. R. 2005. "Project Retrospectives: Evaluating Project Success, Failure, and Everything in Between," *MIS Quarterly* (4:3), pp 361-372.

Newig, J., and Fritsch, O. 2009, "The Case Survey Method and Applications in Political Science". APSA 2009 Toronto Meeting Paper. Available at SSRN: http://ssrn.com/abstract=1451643

Nicholas, J., and Hidding, G. 2010. "Management Principles Associated With IT Project Success," *International Journal of Management and Information Systems* (14:5), pp 147-156.

Nidumolu, S. R. 1995. "The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable," *Information Systems Research* (6:3), pp 191-219.

Nidumolu, S. R. 1996. "A Comparison Of Structural Contingency And Risk-Based Perspectives On Coordination In Software Development Projects," *Journal of Management Information Systems* (13:2), pp 77-113.

Nidumolu, S. R., and Subramani, M. 2003. "The Matrix of Control: Combining Process and Structure Approaches to Managing Software Development," *Journal of Management Information Systems* (20:3), pp 159-197.

Ojiako, U., Johansen, E., and Greenwood, D. 2008. "A qualitative re-construction of project measurement criteria," *Industrial Management + Data Systems* (108:3), pp 405-417.

Ostrom, E. 1989. "Microconstitutional Change in Multiconstitutional Political System," *Rationality and Society, New York: Cambridge University Press.* (1:1), pp 11-50.

Pan, G., Hackney, R., and Pan, S. L. 2008. "Information Systems implementation failure: Insights from prism," *International Journal of Information Management* (28:4), pp 259-269.

Pant, I., and Baroudi, B. 2008. "Project management education: The human skills imperative," *International Journal of Project Management* (26:3), pp 124–128.

Partleton, B. 2011. "PMI 2011 Annual Report: Delivering Value," in *PMI*, PMI.

Peffers, K., Tuunanen, T., Rothenberger, M., and Chatterjee, S. 2007. "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems* (24:3), pp 45-77.

Pereira, J., Cerpa, N., Verner, J., Rivas, M., and Procaccino, J. D. 2008. "What do software practitioners really think about project success: A cross-cultural comparison," *The Journal of Systems and Software* (81:6), pp 897-907.

Petter, S., Straub, D., and Rai, A. 2007. "Specifying Formative Constructs in Information Systems Research," *MIS Quarterly* (31:4), p 623.

PMI, S. c. 2004. *A guide to project management (PMBOK).* (2004 ed.) Project Management Institute: Newton square, PA, USA.

Polites, G. L., Roberts, N., and Thatcher, J. 2012. "Conceptualizing models using multidimensional constructs: a review and guidelines for their use," *European Journal of Information Systems* (21:1), pp 22-48.

Powell, P. L., and Klein, J. H. 1996. "Risk management for information systems development," *Journal of Information Technology* (11:4), pp 309-319.

Pries-Heje, J., and Baskerville, R. 2008. "The Design Theory Nexus," *MIS Quarterly* (32:4), pp 731-755.

Procaccino, D. J., Verner, J. M., Darter, M. E., and Amadio, W. J. 2005. "Toward predicting software development success from the perspective of practitioners: an exploratory Bayesian model," *Journal of Information Technology* (20), pp 187–200.

Purvis, R. L., Sambamurthy, V., and Zmud, R. W. 2001. "The Assimilation of Knowledge Platforms in Organizations: An Empirical Investigation," *Organization Science* (12:2), pp 117-135.

Ragin, C. C. 2009. *Redesigning Social Inquiry: Fuzzy Sets and Beyond*, (1st ed.) University of Chicago Press.

Ravichandran, T., and Arun, R. 2000. "Quality management in systems development: An organizational system perspective," *MIS Quarterly* (24:3), pp 381-415.

Ravichandran, T., and Rai, A. 2000. "Quality management in systems development: An organizational system perspective," *MIS Quarterly* (24:3), p 381.

Ravishankar, M. N., Pan, S. L., and Leidner, D. E. 2011. "Examining the strategic alignment and implementation success of a KMS: A subculture-based multilevel analysis," *Information Systems Research* (22:1), pp 39-59.

Reich, B. H. 2007. "Managing Knowledge and Learning in IT Projects: A Conceptual Framework and Guidelines For Practice " *Project Management Journal* (38:2), pp 5-17.

Ringle, C. M., Sarstedt, M., and Straub, D. W. 2012. "Editor's comments: a critical look at the use of PLS-SEM in MIS quarterly," *MIS Q.* (36:1), pp iii-xiv.

Rivard, S., and Lapointe, L. 2012. "Information Technology Implementers' Responses to User Resistance: Nature and Effects," *MIS Quarterly* (36:3), pp 897-920.

Robey, D., Im, G., and Wareham, J. D. 2008. "Theoretical Foundations of Empirical Research on Interorganizational Systems: Assessing Past Contributions and Guiding Future Directions," *Journal of the Association for Information Systems* (9:9), pp 497-518.

Robey, D., Smith, L. A., and Vijayasarathy, L. R. 1993. "Perceptions of conflict and success in information systems development projects," *Journal of Management Information Systems* (10:1), p 123.

Ropponen, J., and Lyytinen, K. 2000. "Components of Software Development Risk: How to Address Them? A Project Manager Survey," *IEEE Transactions on Software Engineering* (26:2), pp 98-111.

Saarinen, T. 1990. "System Development Methodology and Project Success: An Assessment of Situational Approaches," *Information & Management* (19:3), pp 183-193.

Sauer, C., Gemino, A., and Reich, B. H. 2007. "The Impact of Size and Volatility on IT Project Performance," *Association for Computing Machinery. Communications of the ACM* (50:11), pp 79-84.

Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. 2001. "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems* (17:4), pp 5-36.

Sharma, R., and Yetton, P. 2007. "The Contingent Effects of Training, Technical Complexity, and Task Interdependence on Successful Information Systems Implementation," *MIS Quarterly* (31:2), pp 219-238.

Shenhar, J. A., Tishler, A., Dvir, D., Lipovetsky, S., and Lechler, T. 2002. "Refining the search for project success factors: A multivariate, typological approach," *R & D Management* (32:2), pp 111-126.

Sherif, K., Zmud, R. W., and Browne, G. J. 2006. "Managing peer-to-peer conflicts in disruptive information technology innovations: the case of software reuse," *MIS Quarterly* (30:2), pp 339-356.

Sicotte, C., and Paré, G. 2010. "Success in Health Information Exchange Projects: Solving the Implementation Puzzle," *Social Science & Medicine* (70:1), pp 1159-1165.

Sillince, J. A. A., and Mouakket, S. 1997. "Varieties of political process during systems development," *Information Systems Research* (8:4), pp 368-397.

Simon, H. A. 1996. *The Science of the Artificial*, (Third ed.) MIT Press: Cambridge, MA.

Standish, G. 2011. "Standish Group Chaos Report, CHAOS Summary 2011," Standish Group.

Straub, D., Boudreau, M.-C., and Gefen, D. 2004. "Validation Guidelines for IS Positivist Research," *Communications of AIS (CAIS)* (13:Aricle 24) April, pp 380-427.

Straub, D. 2012. "Editor's comments: does MIS have native theories?," *MIS Quarterly* (36:2), pp III-XII.

Sumner, M. 2000. "Risk factors in enterprise-wide/ERP projects," *Journal of Information Technology* (15:4), p 317.

Taylor, H. 2007. "Outsourced IT Projects from the Vendor Perspective: Different Goals, Different Risks," *Journal of Global Information Management* (15:2), pp 1-27.

Tesch, D., Kloppenborg, J. T., and Frolick, N. M. 2007. " IT Project Risk Factors: The Project Management Professionals Perspective," *The Journal of Computer Information Systems* (47:4), p 61.

Thite, M. 2000. "Leadership styles in information technology projects," *International Journal of Project Management* (18:4), pp 235-241.

Thomas, G., and Fernández, W. 2008. "Success in IT projects: A matter of definition?," *International Journal of Project Management* (26:7), pp 733-742.

Thong, J. Y. L., Yap, C.-S., and Raman, K. S. 1996. "Top management support, external expertise and information systems implementation in small businesses," *Information Systems Research* (7:2), pp 248-267.

Tiwana, A. 2012. "Novelty-Knowledge Alignment: A Theory of Design Convergence in Systems Development," *Journal of Management Information Systems* (29:1) Summer2012, pp 15-52.

Van Bon, J., de Jong, A., and Kolthof, A. 2007. *Foundations of IT Service Management Based on ITIL V3: an introduction*, (3rd ed.) Van Haren Publishing.

Venkatraman, N. 1989. " The Concept of Fit in Strategy Research: Toward Verbal and Statistical Correspondence," *Academy of Management. The Academy of Management Review* (14:3), pp 423-444.

Vinzi, V. E., Chin, W. W., Henseler, J., and Wang, H. 2010. "Editorial: Perspectives on Partial Least Squares.," in *Handbook of Partial Least Squares: Concepts, Methods and Applications.,* S. H. o. C. S. Series (ed.), Springer Handbooks of Computational Statistics Series: Heidelberg/Dordrecht/London/New York: Springer., pp. 1-20.

Wagner, E. L., and Newell, S. 2007. "Exploring the Importance of Participation in the Post-Implementation Period of an ES Project: A Neglected Area," *Journal of the Association for Information Systems* (8:19), pp 508-524.

Wallace, L., Keil, M., and Rai, A. 2004a. "How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model*," *Decision Sciences* (35:2), pp 289-321.

Wallace, L. G., and Keil, M. 2004. "Software Project Risks and Their Effect on Outcomes," *Communications of the Association for Information Systems* (47:4), pp 68-73.

Wallace, L. G., Keil, M., and Rai, A. 2004b. "Understanding software project risk: a cluster analysis," *Information and Management* (42:1), pp 115-125.

Walls, J. G., Widemeyer, G. R., and ElSawi, O. A. 1992. "Building an information system design theory for vigilant EIS," *Information Systems Research* (3:1), pp 36–59.

Wang, T. G. E., Chen, H. G. H., Jiang, J. J., and Klein, G. 2005. "Interaction quality between IS professionals and users: impacting conflict and project performance," *Journal of Information Science* (31:4), pp 273-283.

Warkentin, M., Moore, R. S., Bekkering, E., and Johnston, A. C. 2009. "Analysis of Systems Development Project Risks: An Integrative Framework," *Database for Advances in Information Systems* (40:2), pp 8-27.

Wastell, D. G. 1996. "The fetish of technique: Methodology as a social defence," *Information Systems Journal* (6:1), pp 25-40.

Wateridge, J. 1998. "How can IS/IT projects be measured for success?," *International Journal of Project Management* (16:1), pp 59-63.

Weber, R. 2003. "Editor's comment: still desperately seeking the IT artifact," *MIS Quarterly* (27:2), pp iii-xi.

Wetzels, M., Odekerken-Schröder, G., and van Oppen, C. 2009. "Using Pls Path Modeling for Assessing Hierarchical Construct Models: Guidelines and Empirical Illustration," *MIS Quarterly* (33:1), p 177.

Willcocks, L. P., Lacity, M. C., and Kern, T. 1999. "Risk mitigation in IT outsourcing strategy revisited: longitudinal case research at LISA," *Journal of Strategic Information Systems* (8:3), pp 285-314.

Winter, R. 2008. "Design Science Research in Europe," *European Journal of Information Systems* (17), pp 470–475.

Wixom, H. B., and Watson, J. H. 2001. "An empirical investigation of the factors affecting data warehousing success," *MIS Quarterly* (25:1), pp 17-41.

Wong, B. Year. "Applying the Software Evaluation Framework 'SEF' to the Software Development Life Cycle," 2003, pp. 281-281.

Wong, C.-S., Law, K. S., and Huang, G.-h. 2008. "On the Importance of Conducting Construct-Level Analysis for Multidimensional Constructs in Theory Development and Testing," *Journal of Management* (34:4), pp 744-764.

Wu, C. H. Year. "Exploring impacts of software development process maturity on project risk," Proceedings of the 2008 IEEE-IEEM2008, pp. 1033-1037.

Wu, C. H., Wang, S., and Fang, K. Year. "Investigating the Relationship between IS Project Risk and Project Performance," Third 2008 International Conference on Convergence and Hybrid Information Technology2008, pp. 100-105.

Xia, W., and Lee, G. 2004. "Grasping The Complexity of IS Development Projects," *Association for Computing Machinery. Communications of the ACM* (47:5), pp 68-74.

Xu, X., Zhang, W., and Barkhi, R. 2010. "IT infrastructure capabilities and IT project success: a development team perspective," *Information Technology and Management* (11:3), pp 123-142.

Yetton, P., Martin, A., Sharma, R., and Johnston, K. 2000. "A model of information systems development project performance," *Information Systems Journal* (10:4), pp 263-289.

Yin, R. K., and Heald, K. A. 1975. "Using the Case Survey Method to Analyze Policy Studies," *Administrative Science Quarterly* (20:3) September, 1975, pp 371-381.

Zmud, R. W. 1980. "Management of Large Software Project Efforts," *MIS Quarterly* (4:2), pp 45-55.

Appendix A: The Components of Residual Risk Dimensions

This appendix presents the components of the four dimensions of El-Masri and Rivard (Essay 2) software project residual risk construct. It also details the components' items that were classified by the researchers as well as by the expert judges.

The Seven Components of Project Risk Traits and their Corresponding Items

1- **User/client Risk:** are traits of the customer organization and end-users associated with their level of knowledge, expertise, and commitment to the successful completion of the software project.

1. Experience of end-users' representatives to define system requirements.
2. Knowledge of end-users' representatives with the application domain.
3. Level of end-users commitment to the project.
4. IT competence of customer organization.
5. Extent of customer's responsibility, ownership, and buy-in of the project and its delivered system(s).
6. Availability of end-users' representatives to support the IT team.

2- **Requirement Risk:** are traits of the software project associated with the availability, clarity, and stability of project requirements and supporting documents.

1. Clarity of system requirements.
2. Availability of system requirements.
3. Degree of stability of system requirements.
4. Availability of documentation (e.g., regarding implemented system or related legacy systems).
5. Level of ambiguity of documentation (e.g., regarding implemented system or related legacy systems).

3- **Team Risk:** are traits of the project team and IT personnel (including the project manager) that are related to the level of knowledge, expertise, ability, and commitment to the successful completion of the software project.

1. IT Personnel's general expertise (work with management, team, and effectively perform tasks).
2. IT personnel's knowledge of the user departments and operations.
3. IT personnel's overall administrative skills.
4. IT personnel's expertise with the information system's business domain.
5. IT personnel's development/ implementation expertise (methodology, support tools, project management tools, and implementation tools).
6. IT personnel's knowledge of customer's country, culture, and language.
7. Compatibility of the composition of the project team with project structure (in regards to competencies, coordination, controls, etc.).
8. Morale and level of commitment of project team members.
9. Clarity of role definitions of team members.
10. Project manager's expertise.

4- **Technology Risk:** are traits of the technological components used, implemented, or acquired which could represent sources of risk to the software project.

1. Project requires the acquisition and installation of new hardware.
2. Project requires the acquisition and installation of new software.
3. Degree of maturity of used technology.
4. Degree of stability and appropriateness of the technological architecture, infrastructure, and networks.
5. Number of links to (integration with or interfacing) existing and future systems.
6. Degree of technical complexity.
7. Feasibility of technological solution.
8. Level of Information System's compliancy with the customer's technical architecture.
9. Criticality of the new system roll-out (ease of reverse to prior system of operations).

5- **Resource Risk:** are project traits associated with the sufficiency and stability of resources (including IT personnel, computer machinery, and established methodologies) required to conduct the project.

1. Sufficiency of required budgetary resources to conduct the project.
2. Sufficiency of required time require (person hours) to conduct the project.
3. Sufficiency of human resources to conduct the project.
4. Turnover rate of IT personnel.
5. Expertise of project's suppliers with the required activities.
6. Level of reliability of target (customer) computer machinery.
7. Level of reliability of development computer machinery.
8. Existence of a project champion.
9. Level of institutionalization of effective project management methodology, structure, and standards.
10. Level of institutionalization of effective project control mechanisms over consultants, vendors, and subcontractors.

6- **Project Complexity risk:** Corresponds to project traits related to the complexity and size of the software project and the associated complexity in coordinating with the project's stakeholder groups.

1. Degree of sophistication (expectations) of end-users.
2. Complex architecture and high number of implementation modules.
3. Diversity of stakeholder groups (and end-user organizational units).
4. The number of external subcontractors.
5. The number of external consultants.
6. Number of external vendors furnishing software or hardware components required for the project.
7. Number of hardware and/or software suppliers involved in the project.
8. Level of coordination complexity (need to share resources, need to subcontract, and so on).
9. Project size (cost, time, staffing level, number of affected parties, number of end-users, and number of hierarchical levels occupied by system users).
10. Degree of application complexity.

11. Degree of conflict within system requirements of various stakeholders' groups (level of agreement/ differences in project goals, deliverables, design, etc.).

7- Business risk: Corresponds to traits of the organizational environment and the suitability of the IT investment for the organization's needs, culture and processes.

1. Degree of stability of the business and organizational environments.
2. Soundness of the project's business case.
3. Degree of alignment between organizational culture and the business process changes required by the new IS.
4. Degree of alignment between adopted IS functionality and the needs of the adopting organization.
5. The degree of change the system brings (to procedures, workflow, structures, and so on).
6. Clarity of expected investment benefits.
7. Level of top management support.

The Five Components of Undesirable Events and their Corresponding Items

1- **Organizational/team Structure Events:** Correspond to unfortunate changes to the organizational and team structure and their commitment to project success.

1. Loss of top management support.
2. Changes in organizational management during the project.
3. Changes occurred in organizational priorities.
4. Resignation(s) of key project member(s).
5. Reallocation of project members.
6. Changes in ownership or senior management during project.

2- **Climate-Related Events:** Correspond to events of conflict, miscommunication and resistance between and within stakeholder groups.

1. End-users' resistance.
2. Strains encountered in team relationships and collaboration.
3. Conflict encountered between project stakeholders.
4. Communication failure between the different project stakeholder groups.

3- **Team Performance Events:** Correspond to technical failures and other events related to the inadequate activities and poor performance of IT personnel.

1. Shortfall encountered in IT Personnel performance/ efficiency.
2. Inadequate design.
3. Scope creep.
4. Gold-plating.
5. Developing the wrong functions and properties.
6. Developing the wrong user interface.
7. Failure to transition and migrate data from legacy system.
8. Technical problems.
9. Failure to integrate/interface new IS with legacy applications and other applications.
10. Failure to redesign business processes.

4- **Project Requirement Related Events:** Correspond to the failures, mishaps and other unfortunate events related to the requirements of the software project.

1. Changes occurred to system requirements or scope, Failure to adequately estimate of scope of work.
2. Failure of project consultants or external vendors to meet the project's external dependencies.
3. Inadequately identifying (misunderstanding) system requirements.
4. Failure to specify performance requirements.
5. Failure to select the appropriate packaged solutions.
6. Failure of the affected business units to handle the extent of change.

5- **Project Performance Events:** Correspond to unfortunate events related to the budgetary and schedule performance of the software project.

1. Escalating project cost.
2. Escalating project time of completion.
3. Changes in project's timetable.

The Seven Components of Risk Management Practices and their Corresponding Items

i. **Planning:** Corresponds to practices that aim at analyzing and planning project tasks and processes as well as estimating the required efforts and resources.

1. Detailed cost and schedule estimation.
2. Analyze project mission.
3. Analyze requirement (and requirement changes) costs and benefits.
4. Pilot system.
5. Divide project tasks into smaller chunks.
6. Outsource.
7. Use Incremental approach.
8. Use evolutionary approach.

9. Use modular approach.
10. Reuse existing software.
11. Develop in-house.
12. Form a steering committee.
13. Identify project stakeholders.
14. Define Clear goals and Objectives.
15. Develop detailed project plan.
16. Conduct feasibility analysis of the proposed solution.

ii. **Internal Integration:** Corresponds to practices internal to the project team that aim at ensuring that the team is integrated and can function efficiently.

1. Provide training programs (including cross-training) to IS Personnel.
2. Hire experienced personnel.
3. Match tasks to personnel skills.
4. Select team members with significant previous work relationships.
5. Hire external consultants or consulting company.
6. Meet frequently with team.
7. Meet frequently with steering committee.
8. Follow systems specification standards.
9. Devise award-fee contracts with consultants and IS personnel.
10. Change External Consultants.
11. Restructure project team.
12. Hire additional IT personnel.
13. Reward/sanction IS personnel.
14. Collocate project team to improve communication.
15. Obtain assistance of the vendor of the implemented packaged solution.
16. Perform teambuilding activities (social events).

iii. **Requirements Control:** Corresponds to practices that aim at limiting the unmanageable growth of user requirements and limiting creeping scope.

1. Scale down/scrub requirements/scope.

2. Avoid requirement changes.
3. Minimize customization (Vanilla).
4. Map information requirements and business processes.
5. Prioritize software requirements.

iv. **Performance Control:** Corresponds to practices that aim at insuring the project is progressing according to plan.

1. Check references of external suppliers or consultants.
2. Inspect externally furnished components.
3. Conduct formal user specification approval.
4. Analyze/test product (QA and Performance Testing).
5. Conduct progress reviews.
6. Conduct project milestone phases review.
7. Add slack resources for contingencies.
8. Implement two identical systems each with mirror disks and data replication between the two.
9. Request additional funding.
10. Commission an external review.
11. Assess existing packaged solutions.
12. Assign project champion.
13. Involve top management (of primary stakeholder groups) in project decision-making.

v. **End-user Training:** Corresponds to practices that aim at providing end users training and other material related to the implemented information system.

1. Develop users’ manuals.
2. Provide adequate training to end-users on implemented IS.
3. Provide end-users with adequate training on requirement specification.

vi. **Change Management:** Corresponds to practices that minimize user resistance and help in the transition of end users to the newly adopted information system.

1. Permit voluntary use.
2. Promote IS (Communicate project information and/or Present Prototype/IS to users and their representatives).
3. reward use (provide incentives).
4. Lessen the extent of change the system brings in terms of work processes and organization.
5. Fire resistors.
6. Coerce resistors to cooperate.
7. Perform formal change management disciplines.
8. Provide incentives to stakeholders and user groups to collaborate and share their knowledge with system developers.
9. Manage expectation of end users.
10. Communicate plan to users.
11. Assess the ability to change of the adopter organization and its end-users.

vii. **End-user Participation:** Corresponds to practices that aim at involving the end-user in the requirement elicitation as well as the software development process to insure that the user expectations are understood and achieved.

1. Involve end-users.
2. Prototype/develop scenarios.
3. Developers/implementers collaborate with users.
4. Involve intermediaries that can bridge the communication gap between end-users and system analysts.
5. Cooperate and negotiate software requirements with various stakeholders
6. Observe end-users.
7. Specify the client's (or end-users) acceptance criteria of the intended IS.
8. Use the JAD approach (Joint application development).

The Nine Expected Outcomes:

1- Adherence to budget.
2- Adherence to schedule.
3- Achieving the client's requirements/scope.
4- Meeting quality standards.
5- Minimum amount of rework during the project.
6- Minimum degree of business disruption.
7- Achieving the organization's objective.
8- Yielding business benefits.
9- Degree of use.

Appendix B: Case Survey Search Criteria and the Selected Articles

Search Criteria

ABS((project* OR ISD OR "Information Systems Development" OR
implementation*) AND
("lesson learned" OR "lesson learnt" OR "lessons learned" OR "lessons learnt"
OR case*))
OR TITLE((project* OR ISD OR "Information Systems Development" OR
implementation*) AND
(experience* OR "lesson learned" OR "lesson learnt"
OR "lessons learned" OR "lessons learnt" OR stud* OR
case* OR example* OR investig* OR demonstrate* OR
analys* OR evidenc*))
AND SU[17]("Information systems" OR "Information
technology" OR "Systems development" OR "Systems
engineering" OR Software* OR Computer*)
AND PUB(<<All 38 Journal names joined with the [OR] operator>>)

ABS((project* OR ISD OR "Information Systems Development" OR
implementation*) AND
("lesson learned" OR "lesson learnt" OR "lessons learned" OR "lessons learnt"
OR case*))
OR TITLE((project* OR ISD OR "Information Systems Development" OR
implementation*) AND
(experience* OR "lesson learned" OR "lesson learnt"
OR "lessons learned" OR "lessons learnt" OR stud* OR
case* OR example* OR investig* OR demonstrate* OR
analys* OR evidenc*))
AND SU[18]("Information systems" OR "Information
technology" OR "Systems development" OR "Systems
engineering" OR Software* OR Computer*)
AND PUB(<<All 38 Journal names joined with the [OR] operator>>)

[17] Only for non IS Journals
[18] Only for non IS Journals

Case Survey References

Agarwal, R., Higgins, C., and Tanniru, M. R. 1991. "Technology Diffusion in a Centralized MIS Environment: Experiences at Carrier," *Information & Management* (20:1), pp 61-70.

Ahn, J.-H., and Skudlark, A. E. 1997. "Resolving conflict of interests in the process of an information system implementation for advanced telecommunication services," *Journal of Information Technology (Routledge, Ltd.)* (12:1), pp 3-13.

Akkermans, H. 2002. "Vicious and virtuous cycles in ERP implementation: A case study of interrelations between critical success factors," *European Journal of Information Systems* (11:1), pp 35-46.

Al-Mashari, M., and Al-Mudimigh, A. 2003. "ERP implementation: Lessons from a case study," *Information Technology & People* (16:1), pp 21-33.

Anderson, D., Howell-Barber, H., Hill, J., Javed, N., Lawler, J., and Zheng, L. 2005. "a study of web services projects in the financial services industry," *Information Systems Management* (22:1) Winter2005, pp 66-76.

Apte, U., Sankar, C. S., Thakur, M., and Turner, J. E. 1990. "Reusability-Based Strategy for Development of Information Systems: Implementation Experience of a Bank," *MIS Quarterly* (14:4), pp 421-433.

Aubert, B. A., Barki, H., Patry, M., and Roy, V. 2008. "A multi-level, multi-theory perspective of information technology implementation," *Information Systems Journal* (18:1), pp 45-72.

Avison, D., and Malaurent, J. 2007. "Impact of cultural differences: A case study of ERP introduction in China," *International Journal of Information Management* (27:5), pp 368-374.

Avital, M., and Vandenbosch, B. 2000. "SAP implementation at Metalica: an organizational drama in two acts," *Journal of Information Technology (Routledge, Ltd.)* (15:3), pp 183-194.

Axtell, C. M., Waterson, P. E., and Clegg, C. W. 1997. "Problems integrating user participation into software development," *International Journal of Human-Computer Studies* (47:2), pp 323-345.

Azad, B., and Faraj, S. 2011. "Social power and information technology implementation: A contentious framing lens," *Information Systems Journal* (21:1), pp 33-61.

Bagranoff, N. A., and Brewer, P. C. 2003. "PMB investments: An enterprise system implementation," *Journal of Information Systems* (17:1), pp 85-115.

Bard, J. F., and Graham, A. 1993. "Developing a data-collection system for PCB assembly: a case study in software engineering," *Engineering Management, IEEE Transactions on* (40:2), pp 191-202.

Barker, T., and Frolick, M. N. 2003. "ERP Implementation Failure: A Case Study," *Information Systems Management* (20:4) Fall 2003, pp 43-49.

Bartis, E., and Mitev, N. 2008. "A multiple narrative approach to information systems failure: a successful system that failed," *European Journal of Information Systems* (17:2), pp 112-124.

Bhattacherjee, A. 1998. "Management of emerging technologies: Experiences and lessons learned at US West," *Information & Management* (33:5), pp 263-272.

Beynon-Davies, P., Mackay, H., and Tudhope, D. 2000. "'It's lots of bits of paper and ticks and post-it notes and things . . .': a case study of a rapid application development project," *Information Systems Journal* (10), pp 195-216.

Boonstra, A., and van Offenbeek, M. 2010. "Towards consistent modes of e-health implementation: Structurational analysis of a telecare programme's limited success," *Information Systems Journal* (20:6), pp 537-561.

Bondarouk, T., and Sikkel, K. 2005. "Explaining IT Implementation Through Group Learning," *Information Resources Management Journal* (18:1), pp 42-60.

Bussen, W., and Myers, M. D. 1997. "Executive Information System Failure: a New Zealand case study," *Journal of Information Technology (Routledge, Ltd.)* (12:2), pp 145-153.

Butler, T. 2003. "An institutional perspective on developing and implementing intranet- and internet-based information systems," *Information Systems Journal* (13:3), pp 209-231.

Bygstad, B. 2003. "The implementation puzzle of CRM systems in knowledge-based organizations," *Information Resources Management Journal* (16:4), pp 33-45.

Carugati, A. 2008. "Information system development activities and inquiring systems: an integrating framework," *European Journal of Information Systems* (17:2), pp 143-155.

Chen, C. C., Law, C., and Yang, S. C. 2009. "Managing ERP Implementation Failure: A Project Management Perspective," *Engineering Management, IEEE Transactions on* (56:1), pp 157-170.

Chu, C., and Smithson, S. 2007. "E-business and organizational change: A structurational approach," *Information Systems Journal* (17:4), pp 369-389.

Chun, M., Montealegre, R., and Griffy-Brown, C. 2007. "Leveraging the Sales Force with Portal Technology at the American Subsidiary of a Japanese Motor Company Case Series (A), (B) and (C)," *Communications of the Association for Information Systems* (20:1), p 7.

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

Davidson, E., and Chiasson, M. 2005. "Contextual influences on technology use mediation: a comparative analysis of electronic medical record systems," *European Journal of Information Systems* (14:1), pp 6-18.

Devadoss, P. R., Pan, S. L., and Huang, J. C. 2003. "Structurational analysis of e-government initiatives: A case study of SCO," *Decision Support Systems* (34:3), pp 253-269.

Dong, L., Neufeld, D., and Higgins, C. 2009. "Top management support of enterprise systems implementations," *Journal of Information Technology* (24:1), pp 55-80.

Drummond, H. 1996. "The politics of risk: trials and tribulations of the Taurus project," *Journal of Information Technology* (11:4), pp 347-357.

Fedorowicz, J., Gogan, J. L., and Culnan, M. J. 2010. "Barriers to Interorganizational Information Sharing in e-Government: A Stakeholder Analysis," *Information Society* (26:5), pp 315-329.

Fitzgerald, G., and Russo, N. L. 2005. "The turnaround of the London Ambulance Service Computer-Aided Despatch system (LASCAD)," *European Journal of Information Systems* (14:3), pp 244-257.

Gosain, S., Lee, Z., and Kim, Y. 2005. "The management of cross-functional inter-dependencies in ERP implementations: emergent coordination patterns," *European Journal of Information Systems* (14:4), pp 371-387.

Gogan, J. L., and Gelinas, U. J., Jr. 2007. "Managing the Internet Payment Platform project," *Journal of Information Technology* (22:4), pp 410-419.

Heeks, R., and Stanforth, C. 2007. "Understanding e-Government project trajectories from an actor-network perspective," *European Journal of Information Systems* (16:2), pp 165-177.

Hee-Woong, K., and Pan, S. L. 2006. "Towards a Process Model of Information Systems Implementation: The Case of Customer Relationship Management (CRM)," *Database for Advances in Information Systems* (37:1), pp 59-76.

Irani, Z., Sharif, A. M., and Love, P. E. D. 2001. "Transforming failure into success through organisational learning: an analysis of a manufacturing information system," *European Journal of Information Systems* (10:1), pp 55-66.

Jensen, C. B. 2006. "The Wireless Nursing Call System: Politics of Discourse, Technology and Dependability in a Pilot Project," *Computer Supported Cooperative Work: The Journal of Collaborative Computing* (15:5/6), pp 419-441.

Joshi, K., and Lauer, T. W. 1999. "Transition and change during the implementation of a computer-based manufacturing process planning system: an analysis using the equity implementation model," *Engineering Management, IEEE Transactions on* (46:4), pp 407-416.

Keil, M. 1995. "Pulling the plug: Software project management and the problem of project escalation," *MIS Quarterly* (19:4), pp 421-447.

Kenneth, J. W., and Kozar, S. A. 1978. "Management's Role In Information System Development Failures: A Case Study," *MIS Quarterly* (2:2), pp 7-16.

Kirsch, L. J. 2004. "Deploying Common Systems Globally: The Dynamics of Control," *Information Systems Research* (15:4), pp 374-395.

Kawalek, P., and Wastell, D. G. 1999. "A case study evaluation of the use of the viable system model in information systems development," *Journal of Database Management* (10:4), pp 24-32.

Kautz, K. 2009. "Improvisation In Information Systems Development Practice," *Journal of Information Technology Case and Application Research* (11:4), pp 30-59.

Lapointe, L., and Rivard, S. 2005. "A Multilevel Model of Resistance to Information Technology Implementation," *MIS Quarterly* (29:3), pp 461-491.

Levine, H. G., and Rossmoore, D. 1993. "Diagnosing the Human Threats to Information Technology Implementation: A Missing Factor in Systems Analysis Illustrated in a Case Study," *Journal of Management Information Systems* (10), pp 55-73.

Loeb, K. A., Rai, A., Ramaprasad, A., and Sharma, S. 1998. "Design, development and implementation of a global information warehouse: a case study at IBM," *Information Systems Journal* (8:4), pp 291-311.

Löfgren, A. 2007. "Mobility In-site: Implementing Mobile Computing in a Construction Enterprise," *Communications of the Association for Information Systems* (20:1), p 3.

Luna-Reyes, L. F., Zhang, J., and Cresswell, A. M. 2005. "Information systems development as emergent socio-technical change: a practice approach," *European Journal of Information Systems* (14:1), pp 93-105.

Lyytinen, K., Newman, M., and Al-muharfi, A.-r. A. 2009. "Institutionalizing enterprise resource planning in the Saudi steel industry: A punctuated socio-technical analysis," *Journal of Information Technology* (24:4), pp 286-304.

Lui, K. M., and Chan, K. C. C. 2008. "Rescuing Troubled Software Projects by Team Transformation: A Case Study With an ERP Project," *Engineering Management, IEEE Transactions on* (55:1), pp 171-184.

Mähring, M., Keil, M., Mathiassen, L., and Pries-Heje, J. 2008. "Making IT Project De-Escalation Happen: An Exploration into Key Roles *," *Journal of the Association for Information Systems* (9:8), pp 462-496.

Mähring, M., and Keil, M. 2008. "Information Technology Project Escalation: A Process Model," *Decision Sciences* (39:2), p 239.

Maldonado, E. A., Maitland, C. F., and Tapia, A. H. 2010. "Collaborative systems development in disaster relief: The impact of multi-level governance," *Information Systems Frontiers* (12:1), pp 9-27.

Montealegre, R., and Keil, M. 2000. "De-escalating Information Technology Projects: Lessons from the Denver International Airport," *MIS Quarterly* (24:3), pp 417-447.

Nah, F. F.-H., and Delgado, S. 2006. "Critical Success Factors For Enterprise Resource Planning Implementation And Upgrade," *Journal of Computer Information Systems* (47), pp 99-113.

Newman, M., and Robey, D. 1992. "A Social Process Model of User-Analyst Relationships," *MIS Quarterly* (16:2), pp 249-266.

Nichols, M. L. 1981. "A Behavioral Analysis for Planning MIS Implementation," *MIS Quarterly* (5:1), pp 57-66.

Irani, Z., and Love, P. E. D. 2000. "The Propagation of Technology Management Taxonomies for Evaluating Investments in Information Systems," *Journal of Management Information Systems* (17:3), pp 161-177.

Newman, M., and Sabherwal, R. 1996. "Determinants of commitment to information systems development: A longitudinal investigation," *MIS Quarterly* (20:1), pp 23-54.

Niehaves, B., Klose, K., and Becker, J. 2006. "Governance Theory Perspectives on IT Consulting Projects: The Case of ERP Implementation," *e-Service Journal* (5:1), pp 5-26.

Nordheim, S. 2006. "Implementing enterprise content management: from evolution through strategy to contradictions out-of-the-box," *European Journal of Information Systems* (15:6), pp 648-663.

Pan, G., and Pan, S. L. 2006. "Examining the coalition dynamics affecting IS project abandonment decision-making," *Decision Support Systems* (42:2), pp 639-655.

Pan, G., Pan, S. L., and Newman, M. 2009. "Managing Information Technology Project Escalation and De-Escalation: An Approach-Avoidance Perspective," *Engineering Management, IEEE Transactions on* (56:1), pp 76-94.

Pan, G., and Pan, S. L. 2011. "Transition to IS Project De-Escalation: An Exploration Into Management Executives' Influence Behaviors," *Engineering Management, IEEE Transactions on* (58:1), pp 109-123.

Pardo, T. A., Cresswell, A. M., Thompson, F., and Zhang, J. 2006. "Knowledge sharing in cross-boundary information system development in the public sector," *Information Technology & Management* (7:4), pp 293-313.

Pramatari, K., Evgeniou, T., and Doukidis, G. 2009. "Implementation of collaborative e-supply-chain initiatives: an initial challenging and final success case from grocery retailing," *Journal of Information Technology* (24:3), pp 269-281.

Ray, P. K., and Ray, S. 2010. "Resource-Constrained Innovation for Emerging Economies: The Case of the Indian Telecommunications Industry," *Engineering Management, IEEE Transactions on* (57:1), pp 144-156.

Randolph, G. 2004. "Use-Cases and Personas: A Case Study in Light-Weight User Interaction Design for Amll Development Projects," *Informing Science* (7), pp 105-116.

Sillince, J. A. A., and Mouakket, S. 1997. "Varieties of political process during systems development," *Information Systems Research* (8:4), pp 368-397.

Silva, L., and Hirschheim, R. 2007. "Fighting Against Windmills: Strategic Information Systems and Organizational Deep Structures," *MIS Quarterly* (31:2), pp 327-354.

Spasser, M. A. 2002. "Realist Activity Theory for Digital Library Evaluation: Conceptual Framework and Case Study," *Computer Supported Cooperative Work: The Journal of Collaborative Computing* (11:1/2), pp 81-110.

Sutanto, J., Kankanhalli, A., Tay, J., Raman, K. S., and Tan, B. C. Y. 2008. "Change Management in Interorganizational Systems for the Public," *Journal of Management Information Systems* (25:3) Winter2008, pp 133-175.

Wastell, D. G. 1996. "The fetish of technique: methodology as a social defence," *Information Systems Journal* (6:1), pp 25-40.

Wei, H.-L., Wang, E., and Pei-Hung, J. 2005. "Understanding misalignment and cascading change of ERP implementation: a stage view of process analysis," *European Journal of Information Systems* (14:4), pp 324-334

Appendix C: Project Risk Assessment Questionnaire

Company Alpha

SECTION 1: CONFIDENTIALITY

The following pages contain a questionnaire that was developed as part of a research project at HEC Montreal entitled: Towards a design theory for a Software Project Risk Management System. The information gathered shall remain strictly confidential. It will be used only to advance knowledge and for the dissemination of the overall results at academic or professional forums. Only the researcher and the thesis director (Mdm. Suzanne Rivard) will have access to the data collected. HEC's confidentiality policies stipulate that all information collected is kept confidential, unless Company Alpha specifically agrees to disclose information to a third party or if such a disclosure is required by law. Anonymity is fully preserved and no information that reveals the participants' identities or the identity of the participating organization will be divulged.

Contact information: HEC Montreal, 3000 Côte-Sainte-Catherine

E-mail: Mazen.el-masri@hec.ca Tel: (514) 340-6000 ext. 2553

Mazen El-Masri (Researcher)	First_Name Last_Name (Project Manager)
Date	Date

SECTION 2: GENERAL DESCRIPTION

(1) Project Name: _______________________

(2) Project sponsor: ______________________ There is no project sponsor ☐

(3) Allocated budget: _____________________ Approved by: _________________

(4) Deadline: _________________________ No specific deadline ☐

(5) Will the information system be implemented by an internal team or outsourced to a consultancy company?________________

(6) Describe the responsibilities of the information system vendor as stated in the contract:___________________________

____

(7) Who, if any, is representing the end-users during the information system requirement's determination? _________________

(8) Describe the customer acceptance criteria?_______________________________________

____

(9) Experience as a project manager

No past experience ☐ less than two years ☐ between 2 and 5 years ☐ between 5 and 10 years ☐ greater than 10 years ☐

(10) Experience as a software project manager

No past experience ☐ less than two years ☐ between 2 and 5 years ☐ between 5 and 10 years ☐ greater than 10 years ☐

(11) Number of team members dedicated to the project:

___________ I.T.
___________ Users
___________ Others

SECTION 3: STAKEHOLDERS

(12) List the different groups that will be using the system:

End-user group	Level of Responsibility in the Project			Projected Intensity of Use		
	High	Average	Low	High	Average	Low
	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

(13) Apart from end-user groups, name the other stakeholder groups (e.g.: suppliers, partners, higher management, project team) that will be involved or affected by the project:

Stakeholder group	Level of Responsibility in the Project		
	High	Average	Low
	[ ]	[ ]	[ ]
	[ ]	[ ]	[ ]
	[ ]	[ ]	[ ]
	[ ]	[ ]	[ ]

(14) Others: ____

(15) Comments: ____

SECTION 4: PROJECT SUCCESS CRITERIA

(16) What are the five most important success criteria? Please classify them according to their importance (1 = most important, 5 = least important)

PROJECT SUCCESS CRITERIA	Importance (between 1 and 5, e.g.: 4.3)	Rank (1 à 5)
The project finishes on time.	[ ]	[ ]
The information system achieves its operational objectives.	[ ]	[ ]
The information system satisfies the end-users requirements.	[ ]	[ ]
The disruption to end-users daily operations is kept to the minimum.	[ ]	[ ]
The information system is used by the end-users.	[ ]	[ ]
The project does not pass the allocated budget.	[ ]	[ ]
The information system meets the anticipated quality standards.	[ ]	[ ]
The information system yields its anticipated business benefits.	[ ]	[ ]
The amount of rework during the project is kept to the minimum.	[ ]	[ ]

(17) If one of the top five measures of project success is that the introduction of the information system to the workspace is to achieve operational objectives, describe those anticipated objectives: ____
____

(18) If one of the top five measures of project success is that the introduction of the information system to the workspace is to realise business benefits, describe those anticipated benefits: ____
____

(19) Other success criteria: ____
____

(20) Comments: ____
____

SECTION 5: PROJECT RISK ASSESSMENT

(21) Please evaluate the following characteristics pertaining to the information system project to be conducted.

End user characteristics	Very Low	Low	Average	High	Very High	Not applicable
End-users' representatives experience in defining system requirements.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
End-users' representatives experience with information systems development (or implementation).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
End users' competence with information technologies.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
End-users representatives' knowledge of the application domain.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
End-users representatives' knowledge of the implemented information system.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
End-users commitment to the project.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Customer's responsibility, buy-in, and ownership of the project and its delivered system(s).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Stakeholder's responsibility, buy-in, and ownership of the project and its delivered system(s).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Availability of end-users' representatives to support the IT team.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of negative attitude of end users.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Authority of user groups to define the system requirements.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Project team characteristics	Very Low	Low	Average	High	Very High	Not applicable
Project team's knowledge of the user departments and operations.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Project team's expertise with the information system's business domain.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Project team's experience with the software development (or implementation) methodology and tools.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Clarity of role definitions of team members and other stakeholder groups.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Commitment of team members to the success of the project.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Morale of project team members.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Project team's general experience with working with management and other team members.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Project team's knowledge of customer's country, culture, and language.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Compatibility of the composition of the project team with project structure in regards to competencies and coordination arrangements.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Compatibility of External Consultants work methods with internal staff.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Project team's knowledge with the Information System to be implemented.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Project Manager's decision making authority.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Software requirements characteristics	**Very Low**	**Low**	**Average**	**High**	**Very High**	**Not applicable**
Availability of system requirements.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Clarity of system requirements.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of stability of system requirements.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Availability of documentation on the implemented system.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Availability of documentation on the legacy systems to integrate.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Clarity of documentation on the implemented system.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Technology characteristics	Very Low	Low	Average	High	Very High	Not applicable
Maturity of used technology.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Stability of the technological architecture, infrastructure, and networks.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of integration or interfacing with existing and future systems.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of technical complexity.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Feasibility of technological solution.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Criticality of the new system roll-out (ease of reverse to prior system of operations).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Level of Information System's compliancy with the customer's technical architecture, infrastructure and networks.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
System and information quality of the implemented information system.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Compatibility of the implemented information system and its data structures with other information systems to be integrated with.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
System and Information Quality of legacy systems to migrate from or integrate with.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Resources	**Very Low**	**Low**	**Average**	**High**	**Very High**	**Not applicable**
Sufficiency of required budgetary resources to conduct the project.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Sufficiency of time required to conduct the project.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Sufficiency of human resources to conduct the project.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Turnover rate of project personnel.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Level of institutionalization of effective project management methodology, structure, and standards.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Level of institutionalization of effective project control mechanisms over consultants, vendors, and subcontractors.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Existence of a project champion.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Expertise of project's suppliers with the required activities.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Project complexity	Very Low	Low	Average	High	Very High	Not applicable
Degree of conflict within system requirements of various stakeholders' groups (level of agreement/ differences in project goals, deliverables and design).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Diversity of stakeholder groups (and end-user organizational units).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
The number of external subcontractors (or participating entities in a joint project).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
The number of external consultants.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Number of external vendors furnishing software or hardware components required for the project.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Number of hardware and/or software suppliers involved in the project.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of application complexity.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Level of coordination complexity between project stakeholders (need to share resources, need to subcontract).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Complexity of the information system's architecture.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Organizational characteristics	Very Low	Low	Average	High	Very High	Not applicable
Level of top management support.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of stability of the business and organizational environments.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of alignment between organizational culture and the business process changes required by the new information system.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Soundness of the project's business case	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
The degree of change, unfavourable to some stakeholders group, the system brings to work procedures, workflow, structures, power, and loss of personnel.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Clarity of expected investment benefits.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of alignment between the functionalities of the adopted information system and the needs of the adopting organization and the associated stakeholders.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Degree of alignment between the functionalities of the adopted information system and the governmental laws policies.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
History of failure of the adopter organization in implementing.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Level of bureaucracy of the client organization.	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
The value that the information system brings to the involved stakeholder groups (suppliers, vendors, user departments, other participating companies).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Level of commitment of collaborating stakeholders (vendors, partners, departments, and clients).	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

(22) Generally, unanticipated changes to project's time table by higher management occurs

Always often sometimes seldom never I do not know

☐ ☐ ☐ ☐ ☐ ☐

(23) Other undesirable events that often occur during our projects include: ______________________________
____

(24) The service quality of the software vendor's was verified and was judged to be

Excellent good fair poor not verified I do not know

☐ ☐ ☐ ☐ ☐ ☐

(31) The system quality and performance of the information system were inspected and were judged to be

Excellent good fair poor not inspected I do not know

☐ ☐ ☐ ☐ ☐ ☐

(32) Additional Information: ____
____

APPENDIX D: Project Risk Evaluation Report

Enterprise Content Management System
Company Alpha

Submitted to:

Mr. First_Name Last_Name, chef de projet

Produced by:

Mazen El-Masri, HEC Montréal

Report Résumé

This report presents a detailed analysis of the risk associated with the ULTIMA Document Management System implementation project. At a meeting held on 11 October 2012 with the project manager, the project coordinator, the business analyst (and champion), the user representative, and the system analyst, an initial assessment of project characteristics was performed. The project team also identified the **five primary project objectives** (Table 3). Using the project description provided by the team, we evaluated the **risk profile** based on seven aspects: complexity, team, organization, resources, software requirements, technology, and users (table 1 below).

Risk Factors	Values
F1. Risk associated with project complexity	0.98/5
F2. Risk associated with the team	2.89/5
F3. Risk associated with the organization	3.54/5
F4. Risk associated with resources	2.98/5
F5. Risk associated with the requirements	2.14/5
F6. Risk associated with the technology	4.29/5
F7. Risk associated with the users	3.50/5

Table 1. The seven risk factors

The risk profile of the project combined with the objectives of the project is the basis for the analysis of project risks. This profile was compared with 82 profiles of software projects previously implemented to identify **adverse events** that may occur during the project if no risk management mechanism was used. We measured the probability of occurrence of these events depending on the risk profile of the project. Five types of potential adverse events were identified (Table 2).

Potential Undesirable Events	Probability
E1. Events related to project climate (communication and collaboration)	100%
E2. Events related to software requirements (ex: changes, misunderstanding)	66%
E3. Events related to team performance (ex: technical problems)	61%
E4. Events related to project performance (ex: going over budget)	77%
E5. Events related to organizational or team structure (ex: reallocation of team members)	31%

Table 2. The potential undesirable events and their probability of occurrence

Subsequently, we evaluated the probability that these events constitute an obstacle to the achievement of the project's objectives if no mitigation mechanisms were implemented. We also determined **management mechanisms** that can be implemented to reduce the likelihood of occurrence of these events and their influence on the probability of not meeting the five project objectives (Table 3). Mechanisms have been grouped under three main strategies (M1 user involvement, M2. Controlling Project Performance and M3. Controlling requirement). They are detailed in this report.

Principal project objectives	Importance	Probability of non respect *(before mechanisms)*	Probability of non respect *(after mechanisms)*
O1. The software must be used by end-users	5.0/5.0	31%	10%
O2. The software must attain the operational objectives	4.0/5.0	52%	18%
O3. The software must respond to users' needs	3.5/5.0	34%	17%
O4. The project must cause minimum disruption to the current activities of users	3.0/5.0	55%	13%
O5. The project must respect the planned schedule.	1.0/5.0	61%	27%

Table 3. Principal project objectives

Overall, we believe that the project has a medium level of risk. The risk is mainly related to the characteristics of the technology and users. Probable adverse events may negatively affect the objectives of the project are mainly related to the climate of the project. Using certain mechanisms described in this report can significantly reduce these events.

Project's Risk Profile

The project's risk profile was evaluated according to the seven aspects: complexity, technology, users, needs, resources, team and organization (Figure 1). Taking into account the profile, we simulated the project assuming no management mechanism would be implemented to identify the types of adverse events that may occur. Five types of potential adverse events were identified either climate-related events, software requirements, to the performance of the team, the performance of the project and the organization and / or structure of the team. Events related to climate include project user resistance, conflicts between stakeholders, strained relationships, frustration, and low morale of the project team. Events related to the performance of the project team include inadequate system design, incorrect configuration and functionality, technical difficulties, and performance problems. Events related to software requirements are frequently changing user needs, misunderstanding of requirements, "scope-creep", mis-estimation, and inadequate design of business processes. Finally, the events related to the performance of the project are delays, changes in the project schedule and project cancellation.

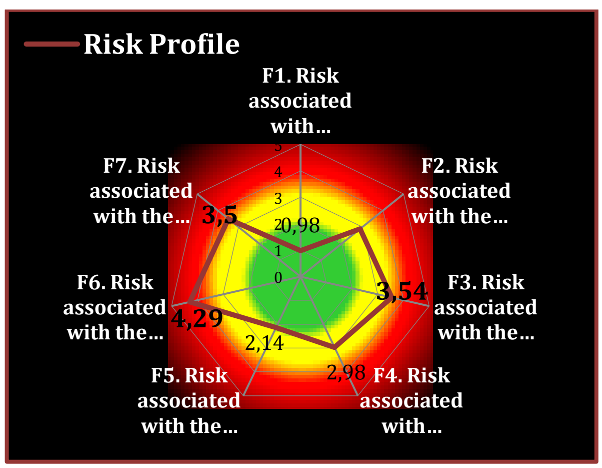


Figure 1: The project's risk profile

Of the five types of events described above, four of them have a probability of occurrence that is between "likely" and "somewhat likely". These are the events related to climate project (E1), the software requirements (E2), the performance of the project team (E3), and project performance (E4) (Figure 2).

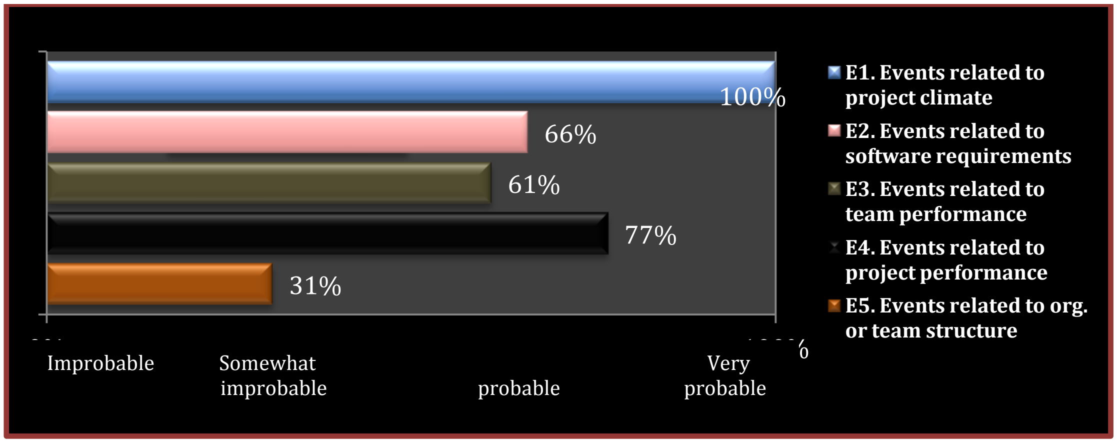


Figure 2: Probable undesirable Events.

The probability of attaining project objectives

Considering the assessment of probabilities of occurrence of adverse events in the previous section, we similarly simulated the project to estimate the probability that the project does not achieve its objectives. For this simulation, we also assumed that no management mechanism would be made. Consequently, the simulation result shows that these adverse events hinder the achievement of operational objectives. It is also quite likely that the project will exceed the expected delivery date and will result in disruptions in current user activities. In addition, there is a likely risk that the project fails as a whole. However, the risk that the package does not meet the needs of users or that it is not used is less likely (see Figure 3).

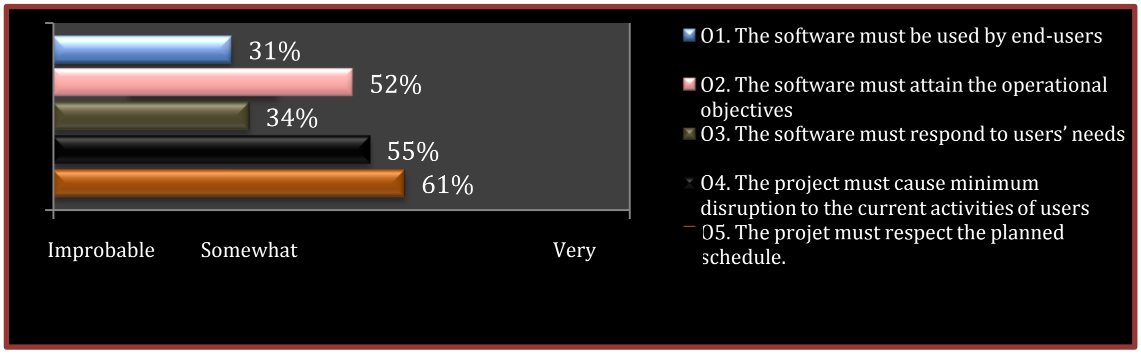


Figure 3: The probabilities of attaining project objectives.

According to the simulation result, events that contribute most to the probability of not achieving the objectives of the project are climate-related events (Figure 4). The main events that may occur are user resistance in relation to the use of software, conflicts between stakeholders, tensions in relations between the project team and users, and frustration and low morale of project team.

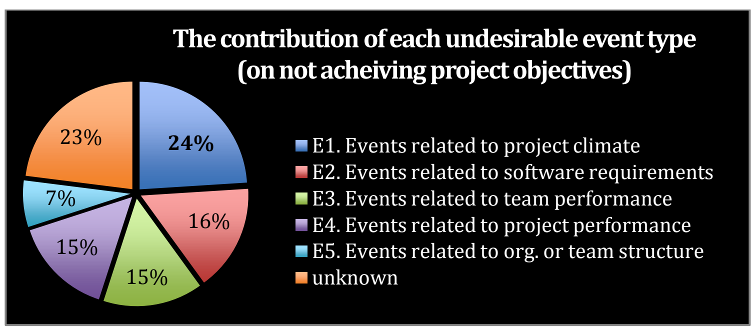


Figure 4: The contribution of undesirable events

Events related to software requirements, to the performance of the team and project performance also contribute to the probability of not achieving the objectives, but to a lesser extent. Events such as the inadequate design of business processes, incorrect configurations, technical difficulties and poor estimation are also events that could occur. We also determined through simulation the contribution of each risk factor that project objectives are not achieved. The results show that, although the risk associated with the technology is the highest (Figure 1), it ranks second in terms of influence on the achievement of project objectives (24%). Indeed, it is the risk associated with users who have the most influence (27%). Risks associated with the organization and the project team have an influence respectively 19% and 15%.

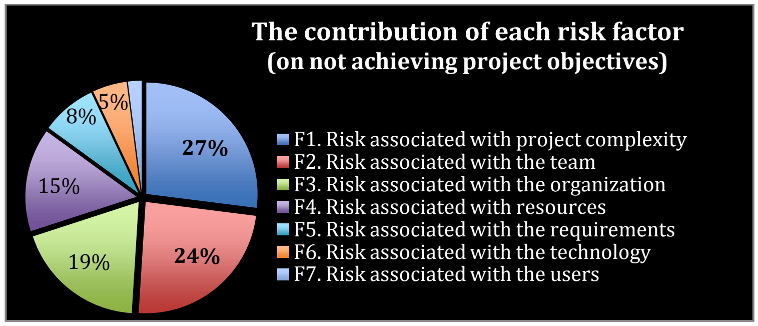


Figure 5: The contribution of risk factors

By combining the probability of not meeting the criteria associated with the severity of not meeting these criteria, we can conclude that the risk exposure of the project is average (Figure 6).

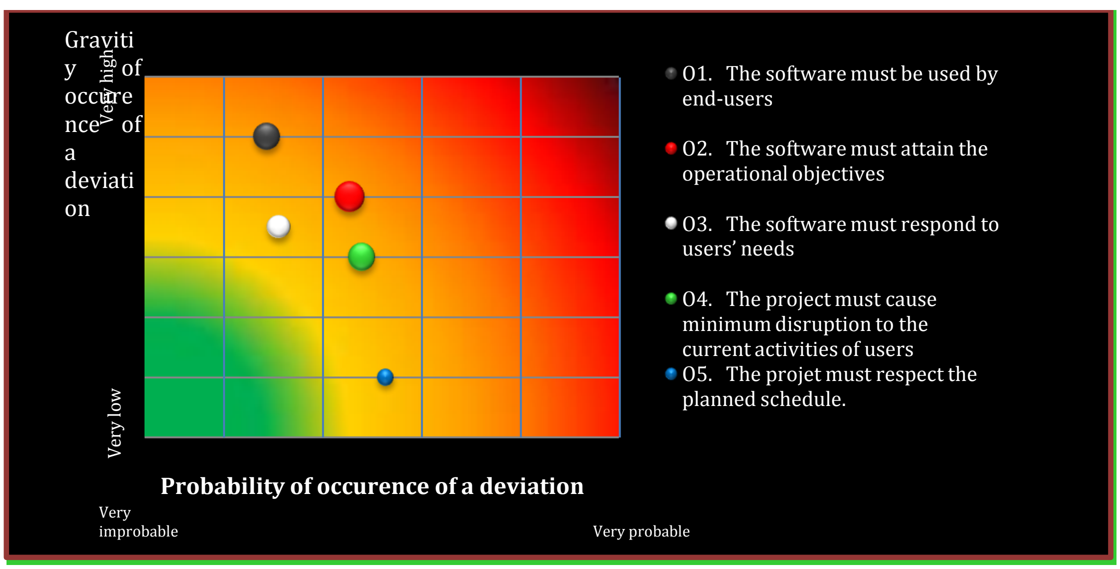


Figure 6: The project risk exposure

Proposed risk management mechanisms

In this section we suggest a list of mechanisms that mitigate project risk by reducing the probability of occurrence of adverse events. Taking into account the specific context of the project, we selected mechanisms from data on 82 projects implementation and we have grouped under three main strategies:

M1. User involvement.

M2. Project performance control.

M3. Requirements control.

M1. User involvement: User involvement helps to reduce the risk of misunderstandings, increases their ownership of the software, and reduces their resistance. There are several techniques suggested to involve users. They are:

(1) Promote the software to users by providing regular information on the planning and progress of the project.
(2) Allow users to use the software prior to launch date and take into account their concerns and ideas.
(3) Define the acceptance criteria with the users.
(4) Develop and revise operational scenarios with the users.
(5) Validate the functionality and business processes with the user representative

M2. Project performance control: Another way to reduce the risk of the project is to monitor closely its performance. There are some techniques that can help you track your project, such as:

(1) Conduct regular assessment of the progress of the project.
(2) Establish benchmarks and milestones.
(3) Select a champion for the project.
(4) Involve senior management in major decisions.

M3. Project requirements control: User requirements should also be checked. Six strategies are suggested:

(1) Prioritize user requirements.
(2) Reduce the scope of user requirements, if necessary.
(3) Avoid changes to requirements established by the users.
(4) Minimize software customization.
(5) Map the software requirements to business processes.

(6) Conduct non-functional requirements testing.

According to our simulation result, the application of the above practices will significantly reduce exposure to risk. Figure 7 below shows the reduction of the overall impact.

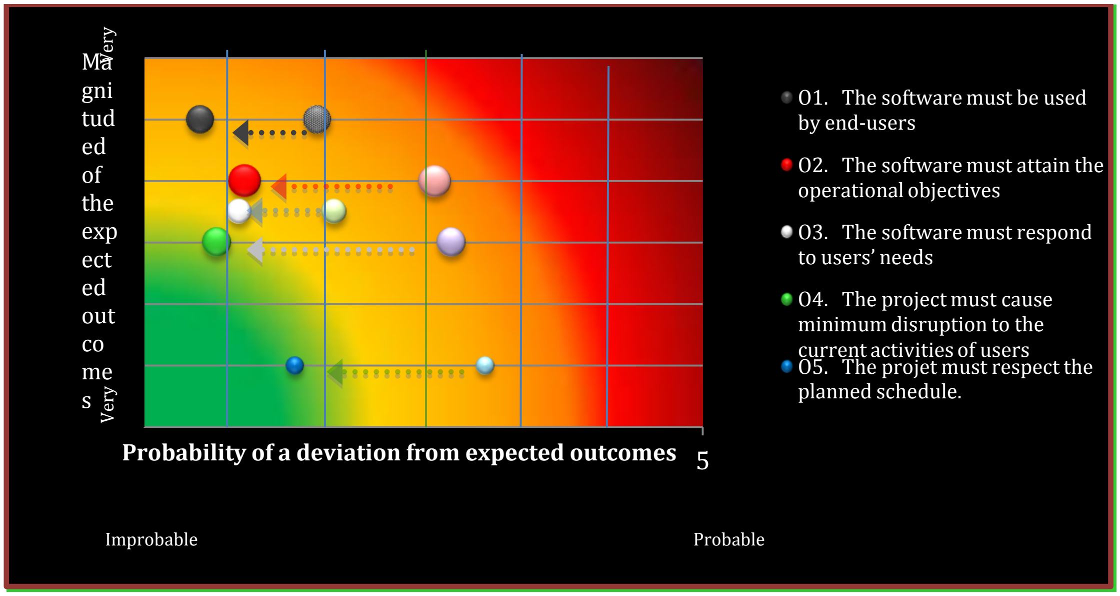


Figure 7: The reduction in risk exposure resulting from the application of the suggested practices.

Concluding Observations

This report summarizes the following:

(1) The risk profile of the project to implement software for document management currently carried out in your organization. This profile is structured into seven risk factors. It includes characteristics of the project collected on October 11 with the project team (Check towards the end of this report for a list of essential characteristics).

(2) The types of adverse events that could occur and their probability of occurrence. To this end, we compared the profile of this project with 82 project profiles previously established and simulated the events that may occur.

(3) The likelihood that the project objectives identified by the project team will not be achieved.

(4) Mechanisms that can be implemented to reduce the likelihood of not achieving the project objectives.

The interpretations of simulation results show that:

(1) The project has an average exposure to risk.
(2) The risk is mainly related to users, technology and organization.
(3) Adverse events that are more likely to occur are related to climate, the requirements and the performance of the team.
(4) None of the main objectives of the project has a high probability of not being attained.
(5) Involving users in the project and monitoring requirements and performance of the team, the risk can be reduced.

Three risk factors require special attention. They are related to the organization, technology and users.

F3. Risk associated with the organization
⇒ Organizational bureaucracy. ⇒ Involvement of stakeholder collaborating on the project. ⇒ Fit between corporate culture and business process changes required by the new system.
F6. Risk associated with the technology
⇒ Maturity of the technology used. ⇒ Quality of software. ⇒ Level of technical complexity. ⇒ Stability of architecture and technological infrastructure and networks. ⇒ Adequacy of software with the technological infrastructure of the customer.
F7. Risk associated with the users
⇒ Users' competence in information technology. ⇒ Users' commitment to the project. ⇒ Users' authority in defining needs. ⇒ Negative attitude of users towards the introduction of new technologies. ⇒ Knowledge of the representatives of the users in regards to the software to implement.