Managing Requirements Risks in IT Projects

Lars Mathiassen
Center for Process Innovation
J. Mack Robinson College of Business
Georgia State University
35 Broad Street, NW, Suite 427
Atlanta GA 30303, USA
phone +1-404-413-7855
lmathiassen@ceprin.org

Tuure Tuunanen¹
University of Oulu
Department of Information Processing Science
P.O. Box 3000
90014 Oulun yliopisto, Finland
phone +358 400 362 501
tuure@tuunanen.fi

¹ Corresponding author
Managing Requirements Risks in IT Project

Abstract: IT professionals must navigate an increasingly complex requirements landscape. Users are often out of immediate reach and new techniques frequently become available. In response, we present a process for managing requirements risks in IT projects. The process helps you analyze risks during the entire project lifecycle, it provides an overview of the toolbox, and it suggests how you can mitigate risks through iterative selection of a suitable portfolio of techniques.

Keywords: requirements risks, management, technique portfolio, project lifecycle.

Introduction
Development of requirements typically starts out from vague ideas that are presented informally and inconsistently and moves cyclically towards agreement on a set of formalized requirements that can serve as a blueprint for IT system design and implementation. Encompassing activities ranging from requirements elicitation and analysis to specification and conflict resolution, requirements development remains a key to IT project success. However, the profile of the discipline has changed in recent years.

Ubiquitous computing, inter-organizational IT systems, and increased introduction of consumer targeted systems [1] have created new challenges. In many IT projects, you don’t know who the users will be, or the users are out of immediate reach. As a consequence, it is difficult to establish the identity of the IT solution and discover its key features. Moreover, a plethora of requirements techniques are available, and new ones are constantly added to the tool box, including a variety of experimental and iterative techniques [2], explication of mental models of users [3], and observation of consumer behavior based on video recording [4].

Our process helps you analyze requirements risks during the entire project lifecycle, it provides an overview of the toolbox, and it suggests how you can mitigate risks through selection of a suitable portfolio of techniques. Integrating state-of-the-art knowledge [5-8] and leveraging previous risk models [9-11], the process is simple and inexpensive to use. If you apply the process iteratively over the whole duration of the IT project lifecycle, it will help you be cost effective, identify requirements that represent needs, and arrive at a complete set of well understood requirements.

The process follows CMMI’s template for risk management: 1) prepare for risk management; 2) identify and analyze risks; and 3) mitigate risks [12]. This simple structure provides you with standardized and easy access to the process and helps you implement it as an integral part of
your organization’s process library and best practices independent of its current process maturity.

**Prepare for Requirements Risk Management**

As a first step of preparation, you must appreciate the types of requirements risks that characterize today’s IT projects. The idea of managing requirements risk is not new [9], but traditional models have become too simplistic. Recent research [5] suggest today’s IT projects face three distinct types of risks:

- **Requirements identity risks**: Many projects develop generic applications or mass-market software to people external to the organization and you may therefore face considerable communication gaps between developers and users. Identity risks are expressions of the physical, conceptual, and cultural distance between you and the users and high risks suggest you don’t known or cannot distinguish key requirements.

- **Requirements volatility risks**: Once you have got access to users’ needs you must consider the stability of requirements. During most IT projects, you are likely to experience changing requirements because stakeholders learn about the solution or because internal or external conditions for IT system usage change. Volatility risks are expressions of how stable requirements are and high risks indicate requirements change easily.

- **Requirements complexity risks**: Once you have arrived at a set of stable requirements, you still have to consider how difficult it is for your team to understand and share requirements. Software is inherently complex and stakeholders’ varying, and often conflicting, views on new systems represent additional sources of complexity. Complexity risks reflect how requirements are conceptualized and structured and high risks indicate requirements are difficult to understand, specify, and communicate.

As a second step of preparation, you must organize an appropriate toolbox for resolving risks. Some techniques are already available within your organization, but you need to evaluate how well they apply to different types of risks. To help you include an appropriate portfolio into your toolbox, we have categorized available techniques into four groups according to their purpose:

- **Requirements discovery techniques**: These techniques are customer-centric and you can use them to identify or predict customer needs. Discovery techniques emphasize the initial identification and understanding of requirements, including those that are tacit to users. Laddering is based on open-ended interviewing of potential users and starts with feature ideas that are further elaborated via questions like “why this is interesting to you” and “how this would work for you”. The aim is to develop a rich understanding of features users would like to have and their reasoning why those features are needed [3]. Video ethnography [4] allows you to go beyond espoused requirements by recording use or interview situations with potential users. In this way, you can get access to studying behaviors and body language of potential users.
• **Requirements prioritization techniques.** These techniques are resource-centric and help you analyze and choose between identified requirements. Using card-sorting [13], you write requirements on physical or virtual cards, which are then rank-ordered by groups of users to create a priority list of requirements. In quality function deployment [14] you use matrixes to iteratively synthesize a priority list of requirements. This technique originates from shipbuilding and has also been successfully used in design of complex products like cars.

• **Requirements experimentation techniques.** These techniques help you evaluate and stabilize requirements through experimentation and by drawing on feedback from end-users. Experimentation techniques are solution-centric and the basic idea is to build a prototype, record comments, and iteratively improve the solution until it has reached a satisfactory level of user acceptance. Contextual design [15] provides you with a portfolio of techniques to understand and receive feedback from the context of use. One example is contextual inquiry, in which you position yourself next to a user in a “master-apprentice” relationship to learn how the user experiences a prototype and to appreciate the context where the IT system eventually will be used.

• **Requirements specification techniques.** These techniques are documentation-centric and based on abstraction of requirements and textual or graphic representation of requirements. Entity-relationship modeling [16] is a classical example that has helped database designers specify complex database solutions for more than three decades. Use cases [17], one of the most used diagramming techniques in the Unified Modeling Language, helps you represent IT system features in relation to specific use situations.

As a final step of preparation, you must share these basic concepts with colleagues and integrate requirements risk management into your organization’s current IT project management practices. Typically, you engage in requirements risk management more than once during a project, allowing adjustments of the adopted approach as the project moves from vague, informal ideas towards a relatively stable and formalized blueprint for system design and implementation. We recommend each session of requirements risk management involves key project stakeholders and lasts hours rather than days.

**Identify and Analyze Requirements Risks**

Generally, you assess risks either quantitatively as the probability of unsatisfactory events multiplied by the loss associated with their outcome, or qualitatively by referring to the uncertainty surrounding your project and the magnitude of potential loss associated with project failure [18]. We suggest a simple, qualitative approach to assess the identity, volatility, and complexity risk profile of developing requirements as summarized in Table 1.

Realizing that the voice of customers and business partners in many IT projects has become important factors in determining requirements, you start by considering identity risks by asking: Who are the users? How are users connected? What is the system context? What are users’ needs? Next, IT projects face several external sources of volatility and the requirements
process is also itself highly dynamic - developers specify requirements, these reveal new options for users and customers, and that leads to new discovery cycles. You can consider volatility risks by asking: How experienced are users? How do users behave? How dynamic is the context? When do requirements change? Finally, requirements can be difficult to comprehend, specify, and communicate and you consider complexity risks by asking: How large is the project? What technology is involved? How diverse are users’ needs? What is the nature of requirements?

### Table 1 Identifying and Analyzing Requirements Risks

<table>
<thead>
<tr>
<th>Risk Area</th>
<th>Risk Question</th>
<th>Low Risk</th>
<th>High Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Identity:</strong></td>
<td>Who are the users?</td>
<td>You know the users and have past experience with them.</td>
<td>You don’t know the users and they are difficult to characterize.</td>
</tr>
<tr>
<td></td>
<td>How are users connected?</td>
<td>You have easy access to users.</td>
<td>You have major difficulties reaching users.</td>
</tr>
<tr>
<td></td>
<td>What is the system context?</td>
<td>You understand the context and the users’ mental models of it.</td>
<td>You are unfamiliar with the context and it might change radically.</td>
</tr>
<tr>
<td></td>
<td>What are users’ needs?</td>
<td>You appreciate users’ needs and have past experience with similar systems.</td>
<td>You don’t know the users’ needs and have no previous experience with similar systems.</td>
</tr>
<tr>
<td><strong>Requirements Volatility:</strong></td>
<td>How experienced are users?</td>
<td>Users are familiar with this type of system.</td>
<td>Users have no experience with this type of system.</td>
</tr>
<tr>
<td></td>
<td>How do users behave?</td>
<td>You know the users’ behavior and it is relatively stable.</td>
<td>You have no experience with similar users and their behavior varies considerably.</td>
</tr>
<tr>
<td></td>
<td>How dynamic is the context?</td>
<td>The context is relatively stable with only a few foreseeable changes.</td>
<td>The context is volatile and future developments are difficult to forecast.</td>
</tr>
<tr>
<td></td>
<td>When do requirements change?</td>
<td>Requirements remain stable over the project life cycle.</td>
<td>Requirements change over the project life cycle.</td>
</tr>
<tr>
<td><strong>Requirements Complexity:</strong></td>
<td>How large is the project?</td>
<td>The project is small with few stakeholders.</td>
<td>The project is large with multiple stakeholders.</td>
</tr>
<tr>
<td></td>
<td>What technology is involved?</td>
<td>The technology is well known within your organization.</td>
<td>The used technology has not been often used or is new to your organization.</td>
</tr>
<tr>
<td></td>
<td>How diverse are users’ needs?</td>
<td>Users have relatively similar needs.</td>
<td>Users have diverse needs that can be hard to prioritize.</td>
</tr>
<tr>
<td></td>
<td>What is the nature of requirements?</td>
<td>Requirements are easy to specify and communicate to stakeholders.</td>
<td>Requirements are difficult to specify and communicate to stakeholders.</td>
</tr>
</tbody>
</table>
For each question, you compare the situation in your project to prototypical low risk and high-risk situations (see Table 1). Subsequently, you synthesize these individual assessments into an overall assessment of each risk area. This process of synthesizing is not a simple question of computing averages across questions. Rather, it is based on discussions in which participants carefully consider all issues and make a qualitative assessment of their overall impact. By systematically enforcing a high-low assessment of each question and each risk area, you end up categorizing your project as having one of only eight possible risk profiles: identity risks (low or high) x volatility risks (low or high) x complexity risks (low or high).

**Mitigate Requirements Risks**

To effectively navigate the complex requirements landscape at any stage of an IT project, you need to select a portfolio of techniques from the toolbox that appropriately addresses the risk profile you face. To that end, you resolve risks following a simple sequence [5]: initially you give primary priority to identity risks (to discover and appreciate requirements), you proceed by primarily emphasizing volatility risks (to evaluate and stabilize requirements), and you conclude by resolving complexity risks (to specify and understand requirements). Hence, depending on the initial risk profile of your IT project, you will progress differently through the eight possible risk profiles as you resolve the risks at hand (see Table 2).

To mitigate risks in any given situation, you must adopt a pattern of techniques that effectively addresses the requirements risk profile you are facing. The underlying principles for how to link resolution patterns to risk profiles are simple and intuitively appealing [5]: 1) resolve identity risks with discovery techniques; 2) resolve volatility risks with prioritization and experimentation techniques; 3) resolve complexity risks with prioritization and specification techniques. Following this logic, Table 2 summarizes what emphasis (high, medium, low) you should put on each type of requirements technique for the eight different risk profiles. As illustrated in the following example, your emphases on the four types of requirements techniques will change as you resolve each of the high requirements risks in your risk profile one at a time.

<table>
<thead>
<tr>
<th>Profile</th>
<th>Identity</th>
<th>Volatility</th>
<th>Complexity</th>
<th>Discovery</th>
<th>Prioritization</th>
<th>Experimentation</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>3</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>4</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>5</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>6</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>7</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>8</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
A Case Study

The Mobile Presence Services (MPS) project was organized within the DiVia\(^2\) R&D program and LTT Research, Inc., a commercial research institute owned by the Helsinki School of Economics. Starting in 2004, the project investigated a new mobile service called ‘mobile presence’ that would attach presence status information to the contact list of a mobile phone. The service would help mobile phone users see if colleagues or friends were available for a chat or voice call. The technology platform was only in beta phase development, and the service had at the time not been sold to any telecom operators.

The goal of the project was to develop requirements for a MPS\(^3\) technology platform based on which mobile phone and service providers could offer a broad range of innovative services to their customers. The project steering group did not give the project specific areas of services to focus on. Instead, the project was tasked to demonstrate how consumers around the globe, in many different ways, might want to use “presence status” information beyond the envisioned basic contact list application.

The MPS project included fifteen IT and mobile technology professionals from four continents and some 450 industry experts and consumers in Auckland, Helsinki, Hong Kong and Las Vegas. The project was initially funded as a joint venture between more than twenty Finnish organizations interested in digital marketing with Nokia as key sponsor. The first version of requirements was delivered in 2006 [19] based on the process described below.

**High identity risks.** To develop a broad range of requirements for mobile presence services, the project began considering: “Who are the users?” and “What is the system context?” Identity risks were considered high, and the project therefore recruited industry experts to help narrow down the scope and key features of mobile presence services. A total of thirteen marketing professionals from participating firms were involved and a group support system (GSS) session was organized in May 2004 to leverage their insights about the mobile services market. The results included recommendations for prototypical scenarios for mobile presence services:

1. A presence enabled mobile service for traveling, which would provide personalized information, such as last minute offers (e.g., flights, events, and hotels) according to the phone user’s profile.

2. A presence enabled mobile service for city dwellers, which would provide information when moving around in downtown areas (e.g., “there are twelve seats available to members of our loyalty program, click here and buy” or “a large ice cream is on offer for $X at Y around the corner”).

---

\(^2\) http://www.divia.fi

\(^3\) http://hdvoice.tmcnet.com/topics/unified-communications/articles/54033-what-mobile-presence.htm
3. A presence enabled mobile service for special interest groups, which would provide relevant personalized information (e.g., a book store announces a new book that matches the phone user’s profile or a retail store announces the arrival of the spring collection to its VIP-customers).

The involved market experts confirmed no similar services were available and mobile phone users therefore had no relevant experiences to inform further requirements development. Instead, the project identified Internet chat services, such as Microsoft Messenger or Skype, as being similar to what was considered here for mobile platforms. As a consequence, the MPS project decided to recruit people who had been among the first to adopt such Internet services as laddering interview participants. The project used snowballing by first identifying a few participants with relevant Internet services experiences, then asking them to recommend peers, and continuing in that fashion until enough participants had been recruited.

Access to this network, allowed the project to conduct nearly 80 laddering interviews in Helsinki, Hong Kong, and Las Vegas to help address “How are users connected?” and “What are their needs?” As participants identified specific services, the interviewer asked, “Why would the service be important to you?” Next, the interviewer would dig deeper with follow-up questions to elicit layers of consequences the participant would expect and communication and information processing requirements the participant valued. These data were systematically documented leading to several thousand requirements and related reasoning for mobile presence services at different locations around the world. Also, laddering interview participants were asked to give a value of one to ten for each requirement and this data was used to prioritize all requirements.

**High volatility risks.** Next, the project assessed volatility risks by asking, “How do users behave?” and “How dynamic is the context?” Volatility risks were considered high due to the dynamic nature of mobile service development and uncertainty on how users would use the MPS. Hence, a decision was made to build an extensive prototyping environment that would allow for collaboration between stakeholders and rapid feedback from users. The project proceeded by developing this environment for MPS prototyping in an advanced CASE tool that supported a wide selection of modeling languages\(^4\). Different iterations of the environment was demonstrated at conferences \([20, 21]\) and development continued until it was ready for industry pilot. At that point in 2005, the project faced an unexpected challenge. Even though the project provided the industry group with a road map for MPS, Nokia decided to shift their technology focus and discontinued the MPS platform.

It was not until 2006 the project found a new way forward. A new team member proposed an Eclipse based development environment would enable the MPS project to generate running prototypes for mobile specific platforms like the MIDP Java and Symbian platforms. The first version of the Eclipse based environment was ready in 2007. The environment allowed the

\(^4\) For further information, see http://www.metacase.com
project to manage several thousand MPS requirements and provide different views of these, for example, graphical requirements maps that provided overviews for managers and drill-down facilities that allowed engineers and analysts access to all interview details. Engineers and analysts could work side by side to design new specific MPS by drawing and connecting different features.

**Low complexity risks.** At this point, the project assessed complexity risks by asking, “How diverse are users’ needs?” and “What is the nature of requirements?” As the focus was placed on a specific set of mobile presence service concepts developed during the previous project phases, the complexity risks were considered low. Hence, the project decided to adopt a version of Unified Modeling Language to specify MPS requirements, thereby transforming select laddering interview requirements to functional and non-functional requirement specifications for automatic code generation and user demonstration [22]. A set of MPS were then developed and run in a test environment with client-server functionality in order to provide proof-of-concept in 2007.

The MPS project revealed key challenges related to development of global IT services. The three different requirements collection locations, Helsinki, Hong Kong and Las Vegas, had distinctly different needs for MPS [19]. Even though the project aggregated the requirements from all three locations into interesting and relevant new MPS concepts, questions were raised whether this approach made sense. Traditionally, mobile phone and service providers would stereotype the global consumer and put all bells and whistles in one “box”, and as a result ignore the specific needs for MPS at different geographic locations around the world. This consideration drove the project to return to analysis of the detailed requirements and leverage the capabilities of the prototyping environment to develop a portfolio of MPS services that represented culturally sensitive requirements.

**Managing Requirements Risks**

As illustrated by the MPS project, requirements risk management helps you proactively identify problems and select appropriate requirements techniques from your toolbox as you move through the IT project lifecycle. The process helps you iteratively identify and analyze important requirements risks (see Table 2), maintain an overview of available techniques, and mitigate identified risks by applying appropriate patterns of techniques (see Table 2). The process is relatively easy to adopt independent on how mature your organization is with respect to processes in general. To adopt the process into your IT projects, we recommend the following steps:

- **Share concepts.** Share the three types of requirements risks and four types of requirements techniques across your project. It is important key stakeholders understand these concepts and are committed to critically analyze and address the project’s requirements risk profile.
• **Update toolbox.** Your requirements toolbox should include key requirement techniques within your organization. Update the toolbox to make sure all four types of techniques are available with appropriate alternative options.

• **Start early.** Initiate requirements risk management during initial project planning. Getting the requirements approach right from the very start will help you establish a sound foundation for requirements development and IT system quality.

• **Repeat later.** Repeat requirements risk management, as initial risks are resolved. This will help you take emerging issues into account and adjust requirements approaches accordingly.

• **Integrate practices.** Addressing requirements risks is essential for IT project success. Still, there are other important types of IT project risks. Requirements risk management therefore needs to be integrated into your project’s comprehensive risk practices.

**References**


Biographies

Lars Mathiassen holds an M. Sc. in computer science from Århus University, Denmark, a Ph.D. in Informatics from Oslo University, Norway, and a Dr. Techn. in software engineering from Aalborg University, Denmark. He is currently a Georgia Research Alliance Eminent Scholar and professor in computer information systems at Georgia State University. His research interests focus on engineering and management of IT-systems. More particularly, he has worked with project management, object-orientation, organizational development, management of IT, and the philosophy of computing. He is a co-author of Professional Systems Development-Experiences, Ideas and Action, Computers in Context-The Philosophy and Practice of Systems Design, Object-Oriented Analysis & Design, and Improving Software Organizations: From Principles to Practice. His research has been published in several international journals including MIS Quarterly, IEEE Transactions on Software Engineering, Information Systems Research, IEEE Transactions on Engineering Management, Information Systems Journal, Communications of the ACM, and IEEE Software.

Tuure Tuunanen is Senior Research Fellow in the Department of Information Processing Science at the University of Oulu. In addition, He is also Adjunct Professor of IS at University of Turku, Associate Professor of IS (Adjunct) at UNLV and a global research fellow at Center for Service Leadership, ASU. He holds a D.Sc. (Econ) in information systems and M.Sc. (Econ) in marketing from Helsinki School of Economics. His current research interests lie in the areas of IS development methods and processes, requirements engineering, risk management, and convergence of IS and marketing disciplines, specifically in design of interactive consumer services and products. His research has been published, e.g., Journal of Management.
Information Systems, Journal of the AIS, Service Sciences, Journal of Information Technology and Theory (JITTA), and Information & Management. Dr. Tuunanen is the Editor-in-Chief of JITTA. Up-to-date information about his research is available at http://www.tuunanen.fi.