Managing Requirements Risks in IT Projects

Lars Mathiassen, *Georgia State University*
Tuure Tuunanen, *University of Oulu*

IT professionals must navigate an increasingly complex requirements landscape. Presented here is a process for identifying, analyzing, and mitigating requirements risks throughout the project life cycle. The authors illustrate its use in a case study for a mobile presence service.

Requirements development typically starts with an informal presentation of vague ideas. It then moves forward cyclically until participants agree on a set of formalized requirements that can serve as a blueprint for IT system design and implementation. The process is key to IT project success and encompasses activities ranging from requirements elicitation and analysis to specification and conflict resolution. However, IT projects have changed in recent years.

Ubiquitous computing, interorganizational IT systems, and an increase in consumer-targeted systems have created new challenges.[1] In many IT projects, the users are either unknown or out of the developer's immediate reach, making it difficult to identify an IT solution. Moreover, a plethora of requirements techniques are available, and new ones are constantly emerging—including techniques that are experimental and iterative,[2] that explicate mental models of users,[3] or that involve observing consumer behavior in video recordings.[4]

Here, we present a process for analyzing requirements risks throughout the entire project life cycle and review its use in a project that investigated requirements for a new mobile service. The process, which integrates state-of-the-art knowledge[5–8] and leverages previous risk models,[9–11] follows the Capability Maturity Model Integration's risk management template of preparation, identification and analysis, and mitigation.[12]

Preparation

Preparing for requirements risk management involves three main steps.

Identify Risk Types

First, you must appreciate the types of requirements risks that characterize today's IT projects. The idea of managing requirements risk isn't new,[9] but traditional models have become too simplistic. Recent research suggest today's IT projects face three distinct types of requirements risks.[5]

Identity. Many projects develop generic applications or mass-market software for people external to the organization, often leading to

 Published by the IEEE Computer Society

considerable communication gaps between developers and users. Identity risks are expressions of the physical, conceptual, and cultural distance between you and the users. A high risk in this area occurs when you don't know or can't identify key requirements.

Volatility. Once you understand the users' needs, you must consider the stability of the requirements. During most IT projects, the requirements change as stakeholders learn more about the solution or as internal or external conditions for IT system usage change. Volatility risks are expressions of requirement stability; high risks indicate that the requirements can easily change.

Complexity. Once you've arrived at set of stable requirements, you still must consider the team's ability to understand and share the requirements. Software is inherently complex, and stakeholders' varying and often conflicting views of new systems represent additional sources of complexity. Complexity risks reflect how requirements are conceptualized and structured; high risks indicate that it's difficult to understand, specify, and communicate the requirements.

Organize a Toolbox

The second preparation step is to organize an appropriate toolbox for resolving risks. Some mitigation techniques are already available in your organization, but you need to evaluate how well they apply to the different risk types. Here, we've categorized available techniques according to their purpose.

Discovery. These customer-centric techniques help identify or predict customer needs, emphasizing the initial identification and understanding of requirements, including those that are tacit to users. For example, *laddering* is an interview technique that asks potential users open-ended questions about feature ideas, then gets users to elaborate by asking follow-up questions such as, "Why is this interesting to you?" and "How would it work for you?" The goal is to develop a rich understanding of the features users would like to have and why.[3]

Another technique, *video ethnography*, lets you go beyond espoused requirements by recording users during an interview or product test.[4] This lets you study user behavior and body language.

Prioritization. These resource-centric techniques help you analyze and choose between identified requirements. *Card sorting* involves writing requirements on physical or virtual cards and having groups of users then rank them, resulting in a prioritized requirements list.[13]

Quality function deployment uses matrices to iteratively synthesize a priority list of requirements.[14] This technique originates from ship building and has also been successfully used to design complex products such as cars.

Experimentation. These solution-centric techniques help you evaluate and stabilize requirements through experimentation and end-user feedback. 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 is a portfolio of techniques for better understanding the use context.[15] For example, *contextual inquiry* creates a "master-apprentice" relationship in which you position yourself close to the user to better understand how he or she experiences the prototype and to learn about the context in which the IT system eventually will be used.

Identity risks are expressions of the physical, conceptual, and cultural distance between you and the users.

Specification. These documentation-centric techniques are based on the abstraction of requirements and their textual or graphic representation. *Entity-relationship modeling* is a classic example—it has helped database designers specify complex database solutions for more than three decades.[16] *Use cases*, a commonly used diagramming technique in the Unified Modeling Language (UML), helps you represent IT system features in relation to specific use situations.[17]

Integrate Risk Management

The final preparation step is to share these basic concepts with colleagues and to integrate requirements risk management into your organization's current IT project management practices.

Table 1. Identifying and analyzing requirements risks.

Risk type	Questions	Low risk	High risk
Requirements identity	Who are the users?	You know the users and have previously worked with them.	You don't know the users; they're difficult to characterize.
	How are users connected?	You have easy access to users.	Users are difficult to reach.
	What is the system context?	You understand the context and the users' mental models of it.	You're unfamiliar with the context, and it could change radically.
	What are users' needs?	You appreciate users' needs and have experience with similar systems.	You don't know the users' needs and have no experience with similar systems.
Requirements volatility	How experienced are users?	Users are familiar with this type of system.	Users have no experience with this type of system.
	How do users behave?	You know the users' behavior; it's relatively stable.	You have no experience with similar users, and their behavior varies considerably.
	How dynamic is the context?	The context is relatively stable with only a few foreseeable changes.	The context is volatile, and future developments are difficult to forecast.
	When do requirements change?	Requirements remain stable over the project life cycle.	Requirements change over the project life cycle.
Requirements complexity	How large is the project?	The project is small with few stakeholders.	The project is large with multiple stakeholders.
	What technology is involved?	The technology is well known within your organization.	The technology hasn't been often used or is new to your organization.
	How diverse are users' needs?	Users have relatively similar needs.	Users have diverse needs that can be difficult to prioritize.
	What is the nature of the requirements?	Requirements are easy to specify and communicate to stakeholders.	Requirements are difficult to specify and communicate to stakeholders.

Typically, you engage in requirements risk management more than once during a project, making adjustments to the adopted approach as the project moves from informal ideas toward a relatively stable and formalized blueprint for system design and implementation. Each session of requirements risk management should involve key project stakeholders and should take a certain number of hours—not days.

Identification and Analysis

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 for developing requirements (see Table 1).

In many IT projects, the voice of customers and business partners is important in determining requirements, so you first want to consider identity risks. Ask who the users will be, how they're connected to the system, what they need, and in what context they plan to use the system.

Next, consider the volatility risks. IT projects face several external sources of volatility, and the requirements process itself is highly dynamic. Developers specify requirements, which reveal new options for users and customers and lead to new discovery cycles. You can consider volatility risks by asking about the user experience and behavior and about the level of stability for the context and requirements.

Finally, requirements can be difficult to comprehend, specify, and communicate, so you need to ask about the project size, technology, diversity of user needs, and nature of the requirements.

For each question, compare the situation in your project to prototypical low-risk and high-risk situations (see Table 1). Then, synthesize these individual assessments into an overall assessment of each risk area. Don't just compute averages across questions; rather, have all participants carefully consider the issues and make a qualitative assessment of the overall impact. By systematically enforcing a high-low assessment of each question and risk area, you'll end up categorizing the project as one of only eight possible risk profiles: identity risks (low or high) × volatility risks (low or high) × complexity risks (low or high).

Table 2. Linking risk profiles to resolution patterns.

	Risk type			Mitigation technique			
Profile	**Identity**	**Volatility**	**Complexity**	**Discovery**	**Prioritization**	**Experimentation**	**Specification**
1	High	High	High	High	Low	Medium	Medium
2	Low	High	High	Low	High	High	Medium
3	High	Low	High	High	Low	Low	Medium
4	High	High	Low	High	Low	Medium	Low
5	Low	Low	High	Low	High	Low	High
6	Low	High	Low	Low	High	High	Low
7	High	Low	Low	High	Low	Low	Low
8	Low	Low	Low	Low	Medium	Low	Low

Mitigation

To effectively navigate the complex requirements landscape at any stage of an IT project, you must select an appropriate portfolio of techniques from the toolbox. To that end, first focus on identity risks (to discover and appreciate requirements), then volatility risks (to evaluate and stabilize requirements), and finally complexity risks (to specify and understand requirements).[5] Depending on the project's initial risk profile, you'll progress differently through the eight possible 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 your requirements risk profile. The underlying principles for how to link resolution patterns to risk profiles are as follows:[5]

- resolve identity risks with discovery techniques,
- resolve volatility risks with prioritization and experimentation techniques, and
- resolve complexity risks with prioritization and specification techniques.

Following this logic, Table 2 summarizes what emphasis (high, medium, or low) you should put on each risk type and mitigation technique for the eight different risk profiles. Your emphasis on the four types of requirements techniques will change as you resolve the high requirements risks, one at a time.

A Case Study

In 2004, the DiVia research and development program (www.divia.fi) and LTT Research, a commercial research institute owned by the Helsinki School of Economics, organized the mobile presence service (MPS) project. The goal was to investigate a new mobile service that attached presence-status information to a mobile phone's contact list. This new service aimed to help cell 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 not yet been sold to any telecom operators.

Project participants had to develop requirements for an MPS technology platform[19] that mobile phone and service providers could use to offer a broad range of innovative services to their customers. The project steering group didn't ask the participants to focus on specific areas of services. Instead, participants were tasked with demonstrating how consumers around the globe might want to use presence-status information beyond the envisioned basic contact list application.

The MPS project included 15 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 among more than 20 Finnish organizations interested in digital marketing, with Nokia as key sponsor. The first version of requirements was delivered in 2006, based on the process we describe here.[20]

Assessing Identity Risks

To develop a broad range of MPS requirements, the project began considering the users and system context. Identity risks were considered high, so participants recruited industry experts to help narrow the project scope and identify key MPS features. In May 2004, they organized a GSS

(group-support-system) session involving 13 marketing professionals from participating firms to leverage their insights about the mobile services market.

The results included recommendations for prototypical scenarios based on the following presence-enabled mobile services:

- provide personalized information for travelers, such as last minute offers (on flights, events, hotels, and so on) according to the phone user's profile;
- provide information to city dwellers moving around downtown, letting them know, for example, that "there are 12 seats left for the show—click here to buy" or that "ice cream is on sale at the shop around the corner"); and
- provide relevant information for special interest groups—for example, a book store might announce a new book that matches the phone user's profile, or a retail store might announce the arrival of its spring collection to its VIP customers.

The mobile presence service (MPS) testing revealed key challenges in developing global IT services.

The market experts confirmed that similar services weren't available, so mobile phone users had no relevant experiences to inform further requirements development.

Instead, the project participants identified Internet chat services, such as Microsoft Messenger or Skype, as similar to what was being considered for mobile platforms. They decided to conduct laddering interviews to obtain more information, using *snowballing* to recruit participants—that is, they identified a few users with relevant Internet services experiences, then asked them to recommend peers, and continued in that fashion until enough they had enough interviewees.

They conducted nearly 80 laddering interviews in Helsinki, Hong Kong, and Las Vegas to help determine user needs and system features. As the interviewees identified specific services, the interviewer would ask, "Why would the service be important to you?" Then, the interviewer would dig deeper to elicit from the interviewee the layers of consequences they expected and the communication and information-processing requirements they valued. The interviewers systematically documented the data, leading to several thousand MPS requirements and related reasoning for the different locations. The interviewees also assigned a numeric value (from 1–10) to each requirement, which helped prioritize them.

Assessing Volatility Risks

Next, project participants assessed volatility risks by asking about user behavior and the level of context stability. They considered the risks to be high owing to the dynamic nature of mobile service development and the uncertainty surrounding MPS use. So, they decided to build an extensive prototyping environment that would allow for collaboration between stakeholders and help obtain rapid feedback from users.

They developed the MPS prototyping environment using an advanced computer-aided software engineering tool that supported a wide selection of modeling languages (www.metacase.com) and demonstrated different iterations of the environment at conferences.[21,22] Development continued until 2005, at which point the prototype was ready for industry pilot. However, Nokia then decided to shift its technology focus and discontinued the MPS platform.

In 2006, the project found a way forward when a new team member proposed using an Eclipse-based development environment to generate running prototypes for mobile-specific platforms, such as the Java and Symbian Mobile Information Device Profile (MIDP) platforms. The first version of the Eclipse-based environment was ready in 2007. The environment let project participants manage and provide different views of several thousand MPS requirements. For example, graphical requirements maps provided overviews for managers, and drill-down facilities gave engineers and analysts access to all interview details. Engineers and analysts could work side by side to design new specific MPS features by drawing and connecting different features.

Assessing Complexity Risks

Participants assessed the complexity risks by asking about the diversity of user needs and the

nature of the requirements. They were focusing on a specific set of MPS concepts developed during previous project phases, so they considered the complexity risks to be low.

They adopted a version of UML to specify MPS requirements, transforming select laddering interview requirements into functional and nonfunctional requirement specifications for automatic code generation and user demonstration by generating software prototypes on the fly.[23] Then, they developed an MPS set and ran it in a test environment with client-server functionality to provide a proof-of-concept in 2007.

The MPS testing revealed key challenges in developing global IT services. The three different requirements-collection locations—Helsinki, Hong Kong, and Las Vegas—had distinctly different MPS needs.[20] Even though the project aggregated the requirements from all three locations into interesting and relevant new MPS concepts, researchers questioned 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," ignoring specific MPS needs based on geographic location. This consideration drove the project participants to further analyze the detailed requirements and leverage the capabilities of the prototyping environment to develop a portfolio of MPS services that represented culturally sensitive requirements.

As the MPS project shows, requirements risk management helps you proactively identify problems and select appropriate requirements techniques as you move through the IT project life cycle. To adopt the process in your IT projects, we recommend the following steps:

- *Share concepts.* Share the three types of requirements risks and four types of requirements mitigation techniques across your project. Key stakeholders must understand these concepts so they can critically analyze and address the project's requirements risk profile.
- *Update the 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 you resolve initial risks. This will help you consider emerging issues 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 must therefore be integrated into your project's comprehensive risk practices.

The process is relatively easy to adopt, independent of your organization's process maturity level, and will help you iteratively identify and analyze important requirements risks, maintain an overview of available techniques, and mitigate identified risks.

Requirements risk management helps you identify problems and select appropriate requirements techniques.

References

1. T. Tuunanen et al., "A Conceptual Framework for Consumer Information Systems Development," *Pacific Asia J. Assoc. for Information Systems,* vol. 2, no. 5, 2010, pp. 47–66.
2. C. Larman and V.R. Basili, "Iterative and Incremental Development: A Brief History," *Computer,* vol. 36, no. 6, 2003, pp. 47–56.
3. K. Peffers and C. Gengler, "How to Identify New High-Payoff Information Systems for the Organization," *Comm. ACM,* vol. 46, no. 1, 2003, pp. 83–88.
4. M. Jirotka and P. Luff, "Supporting Requirements with Video-Based Analysis," *IEEE Software,* vol. 23, no. 3, 2006, pp. 42–44.
5. L. Mathiassen et al., "A Contigency Model for Requirements Development," *J. Assoc. of Information Systems,* Nov. 2007, pp. 569–597.

6. B. Boehm, "Software Risk Management: Principles and Practices," *IEEE Software,* vol. 8, no. 1, 1991, pp. 32–41.
7. K. Lyytinen et al., "Attention Shaping and Software Risk—A Categorical Analysis of Four Classical Risk Management Approaches," *Information Systems Research,* Sept. 1998, pp. 233–255.
8. L. Mathiassen et al., "A Contingency Model for Requirements Development," *J. Assoc. Information Systems,* Nov. 2007, pp. 569–597.
9. G. Davis, "Strategies for Information Requirements Determination," *IBM Systems J.,* vol. 21, no. 1, 1982, pp. 4–31.
10. B. Fazlollahi and M.R. Tanniru, "Selecting a Requirement Determination Methodology-Contingency Approach Revisited," *Information & Management,* Dec. 1991, pp. 291–303.
11. A.M. Hickey and A. Davis, "A Unified Model of Requirements Elicitation," *J. Management Information Systems,* vol. 20, no. 4, 2004, pp. 65–84.
12. *CMMI for Development, Version 1.2*, tech. report CMU/SEI-2006-TR-008, Software Eng. Inst., Carnegie Mellon Univ., 2006; www.sei.cmu.edu/reports/06tr008.pdf.
13. T.A. Byrd et al., "A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques," *MIS Quarterly,* Mar. 1992, pp. 117–138.
14. T. Ravichandran and A. Rai, "Quality Management in Systems Development: An Organizational System Perspective," *MIS Quarterly,* Sept. 2000, pp. 381–415.
15. K. Holtzblatt, "Requirements Gathering: The Human Factor," *Comm. ACM,* vol. 38, no. 5, 1995, pp. 31–33.
16. P.P.-S. Chen, "The Entity-Relationship Model—Toward a Unified View of Data," *ACM Trans. Database Systems,* Mar. 1976, pp. 9–36.
17. G. Booch et al., *The Unified Modeling Language User Guide*, 1st ed., Addison Wesley, 1998.
18. H. Barki et al., "Toward an Assessment of Software Development Risk," *J. Management Information Systems,* vol. 10, no. 2, 1993, pp. 203–225.
19. R. Watson, "What Is Mobile Presence?" TMCnet.com, 10 Apr. 2009; http://hdvoice.tmcnet.com/topics/unified-communications/articles/54033-what-mobile-presence.htm.
20. T. Tuunanen et al., "Developing Feature Sets for Geographically Diverse External End Users: A Call for Value-Based Preference Modelling," *J. Information Technology Theory and Application,* vol. 8, no 2, 2006, pp. 41–55.
21. M. Rossi and T. Tuunanen, "A Method and Tool for Wide Audience Requirements Elicitation and Rapid Prototyping for Mobile Systems," *Conceptual Modeling for Advanced Application Domains*, LNCS 3289, Springer, 2004, pp. 629–640.
22. T. Tuunanen and M. Rossi, "Engineering a Method for Wide Audience Requirements Elicitation and Integrating It to Software Development," *Proc. 37th Hawaii Int'l Conf. System Sciences*, IEEE CS Press, 2004, p. 10.
23. M. Przybilski and T. Tuunanen, "From Rich User Requirements to System Requirements," *Proc. 11th Pacific-Asia Conf. Information Systems*, Association for Information Systems (AIS), 2007, pp. 561–574.

Lars Mathiassen *is 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—in particular, project management, object orientation, organizational development, and the philosophy of computing. Mathiassen received his PhD in informatics from Oslo University, Norway, and a Dr. Techn. in software engineering from Aalborg University, Denmark. Contact him at lmathiassen@ceprin.org.*

Tuure Tuunanen *is a senior research fellow in the Department of Information Processing Science at the University of Oulu. He's also an adjunct professor of information systems at the University of Turku, an associate professor of information systems (adjunct) at the University of Nevada, Las Vegas, and a global research fellow at the Arizona State University's Center for Service Leadership. His research interests include information systems development methods and processes, requirements engineering, risk management, and the convergence of information systems and marketing disciplines, specifically in the design of interactive consumer services and products. Tuunanen received his PhD in information systems from the Helsinki School of Economics. Contact him at tuure@tuunanen.fi or www.tuunanen.fi.*

Computer DIGITAL

The IEEE Computer Society's next-generation flagship publication

More value, more content, more resources

For computing professionals, keeping abreast of the industry's most exciting developments is a continuous process. Beginning in January, *Computer Digital* will offer you even more tools to accomplish that at no risk.

Computer Digital will deliver the same great peer-reviewed articles and columns as the print version of *Computer*. PLUS, it will be:

Mobile—Read the email-delivered issue anytime, anywhere, at your convenience—on your laptop, iPad, or other mobile device.

Searchable—Quickly find the latest information in your fields of interest. Access the digital archives, and save what's most relevant to you.

Linked—Click on table of contents links and instantly go to the articles you want to read first. Article links go to additional references to deepen your new discoveries.

Engaging—Absorb the latest high-tech concepts as they come to life through related audio, video, animation, and more. Email authors directly. Even apply for jobs through convenient ad links.

View Demo Now
computer.org/computer-demo

Current digital subscribers—Look for the *Computer Digital* January issue link coming your way in December.

Current print subscribers—Switch from print to digital by 8 December 2011 to receive the January issue. See link below.

Make the switch at computer.org/computerdigital

IEEE

IEEE computer society