



**This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.**

**Author(s):** Tuunanen, Tuure; Vartiainen, Tero; Ebrahim, Mehdi; Liang, Murong

**Title:** Continuous Requirements Risk Profiling in Information Systems Development

**Year:** 2015

**Version:** Published version

**Copyright:** © 2015 IEEE

**Rights:** In Copyright

**Rights url:** <http://rightsstatements.org/page/InC/1.0/?language=en>

**Please cite the original version:**

Tuunanen, T., Vartiainen, T., Ebrahim, M., & Liang, M. (2015). Continuous Requirements Risk Profiling in Information Systems Development. In T. X. Bui, & R. H. Sprague (Eds.), Proceedings of the 48th Hawaii International Conference on System Sciences (HICSS 2015) (pp. 4019-4028). IEEE. Proceedings of the Annual Hawaii International Conference on System Sciences.  
<https://doi.org/10.1109/HICSS.2015.483>

# Continuous Requirements Risk Profiling in Information Systems Development

Tuure Tuunanen  
University of Jyväskylä  
tuure@tuunanen.fi

Tero Vartiainen  
University of Oulu  
tero.vartiainen@oulu.fi

Mehdi Ebrahim  
University of  
Auckland

Murong Liang  
University of  
Auckland

## Abstract

*With the increasing adoption of agile, lean, and iterative development methods, information systems development (ISD) has become continuous, meaning that system development moves rapidly from release to release. This means that work practices and challenges that practitioners face have changed. Despite these changes, requirements development is still critical in ISD. However, IS literature is silent on how to manage requirements-related risks in the practice of continuous IS development. To fill this gap, we propose a continuous requirements risk profiling method. The study is informed by design science research methodology, and we apply focus group interviews and a Delphi study for data collection to support the method development. The developed method can be integrated to ISD projects using different continuous ISD methods.*

## 1. Introduction

Today's ISD development practices can be characterized as continuous. However, this change has not impacted the way that risk management for ISD—specifically requirements development—is viewed in the literature. Nor does practice offer solutions of how do to requirements risk profiling in a continuous manner. This paper applies design science research methodology to develop a continuous requirements profiling method to fill this gap in the literature and practice. Below, we provide an argument for the need for this method, our research objective and question, and finally how we set to accomplish this task.

During the 1970s and 1980s, structured methods such as Waterfall [1] were typical. However, as the pace of business became more fast and demanding, the ISD methods had to respond to the changes. During the 1990s, there was a movement from plan-driven methods toward more agile methods [2] that are argued to be more market-driven and user-driven [3-4]. Short cycle times and quick releases are typical in agile. The emphasis on release orientation and parallel development means that ISD does not begin or end; rather, there is continuous development from release to release [5]. Indeed, the idea of continuous development

is visible in continuous integration [6-8], which means that members of a software development team integrate their work frequently—daily, for example. Also, users are integrated in continuous software development through error reporting systems [9].

Requirements risks have been studied from different aspects. In the early 1980s, the focus was on developing an understanding of how to manage project portfolios and how and when we should use different requirements' engineering techniques [10-13]. Later on, in the 1990s, researchers examined what risk development projects entail [14-15] and which of these are more important at certain phases of the development project [16]. More recently, the focus has shifted to the questions of how to assist the practitioners to navigate the requirements engineering landscape; see, e.g., [17].

However, in our understanding, until our study, there have not been specific methods available in the literature or practice that would offer a continuous approach to requirements risk profiling [10-17]. The most recent notable studies have also been either literature review-driven [18] or conceptual [17]. Our study differs in this respect, as we follow design science research methodology [19] to develop a requirements risk profiling method in cooperation with the Project Management Institute of New Zealand. More specifically, we engaged practitioners in the method development process by applying 1) focus group interviews and 2) a Delphi survey with industry experts. With the selected research approach, we seek to answer how continuous requirements risk profiling can be accomplished in an IS development project.

The paper is structured as follows. We first review the literature on continuous development and requirements risk profiling. Thereafter, we depict the chosen research methodology more in detail and report the method development and evaluate the method. Finally, we discuss the implications of our work and conclude.

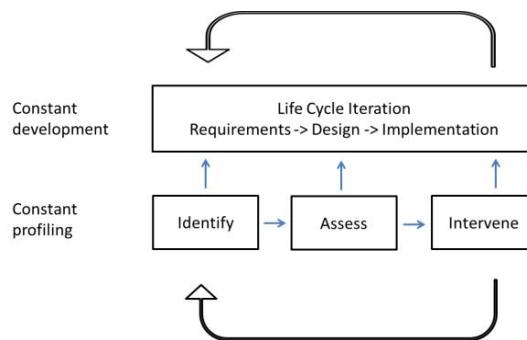
## 2. Continuous Development

Today's ISD development can be characterized as constant development and constant profiling (Figure 1). Firstly, constant development is characteristic in emergent organizations where there is a constant need to change due to the rapid changes in business environments [20]: In fact, ISs do not follow a

traditional life cycle that terminates the IS; instead, the ISs are preserved and continuously redeveloped to satisfy users' needs. Characteristic in emergent organizations is that users are constantly unsatisfied with the ISs, and if they were satisfied, it would be a sign of failure. This means that traditional ISD values, such as a long IT system lifespan, concise specifications, and a complete systems analysis, are outmoded. Instead, there should be an emphasis on continuous analysis, negotiated requirements, and continuous maintenance activities [20].

Secondly, constant profiling means that continuously changing variables in the ISD process are constantly observed and assessed (cf. [21] on the differences between dynamic and discrete models in software engineering). As an example, Mathiassen et al. [18] propose that requirements risks should be assessed continuously with appropriate intervals (cf. notion of continuous risk management in [16]). As another example, users are integrated in continuous software development through error reporting systems [9]. These examples show that there is constant identification, assessment, and intervention in emerging problems.

All lifecycle models involve some form of requirements gathering or system specification phase, a build phase, and an implementation phase where the systems are handed over to the clients [22]. In constant development (using, e.g., agile and lean) these phases iterate numerous times in short intervals. Therefore, risk identification, assessment, and intervention repeat, respectively (Figure 1).



**Figure 1. Continuous development**

### 3. Requirements Risk Profiling in ISD

Requirements risk management typically starts with efforts to identify the risks that projects can be exposed to. There have been numerous studies that have put forward lists of risk items that projects are exposed to. Schmidt et al. [23] conducted a Delphi study that resulted in the development of a ranked risk list affecting projects. Such studies have also been conducted earlier, which have identified risk items affecting projects [24]. Requirements risk management has been also seen as a way to control and mitigate the risks that projects are exposed to. Just as with IS

development lifecycles, which follow a more or less iterative nature to development [1, 25-27, 29], for risk management to be effective, it needs to be iterative, as well. An iterative approach to risk management will include a form of monitoring risks and the severity that the risk item possesses to the overall project [30]. Conrow and Shishido [30] showed how an extended version of Boehm risk management [31] was used to manage the risks in software-intensive projects. A task that needs to be performed to manage risks is to assess the risk exposure of projects.

Barki et al. [14] proposed a software risk measurement method, which incorporated the uncertainty that projects were exposed to and the impact that project failure can have. This study moved from measuring the probability of an unexpected event [31] to measuring the uncertainty in the projects. Both of the studies attempted to provide a tool to measure the overall risk exposure of the project. Boehm [31] introduced risk management as a two-step process involving assessment and management, which involved identification, prioritization, resolution monitoring, etc.

While Barki et al. [14] and Boehm [31] proposed a risk measurement tool for the overall projects, their research did not specify how to interpret the resulting risk exposure. Further research was required. Tiwana and Keil [32] proposed a one-minute measurement tool that would allow project managers to determine the current risk exposure of the project varying from high risk to low risk. The measurement tool was based on six risk drivers that were determined from the examination of 720 projects. Similar to Keil et al. [24], the risk drivers were classified based on the level of control that project managers had on them. Wallace [33] conducted a study that identified six main categories of risk that can affect projects. Further, Wallace et al. [34] described how the same six categories of risks varied for different types of projects, such as high-, medium-, and low-risk projects. Their focus, however, was on three main project characteristics: scope, strategic orientation, and in-house versus outsourced projects. Their results indicated that both strategic projects and outsourced projects had significant risk exposure and, therefore, required careful management.

The latest in the area of work focusing on risk management in project management [10-13] and risk measurement is Mathiassen et al. [18]'s Contingency Model for Requirements Development. Their focus is on the requirements of ISD projects. For ISD projects, requirements play a crucial part [22]. This also should be the driver for the measurement of project risks. The model proposed by Mathiassen et al. [18] focused on requirements by categorizing them into identity (the availability of requirements) and volatility (the stability of requirements and complexity of requirements). Their model specified that the risk level (high, medium, or low) of a project is based on the type of requirement risk that the project faced. However, the model

proposed in their study is conceptual and is based on the literature review.

Mathiassen et al. [18] characterize risks related to **requirements identity** for situations in which there is a communication gap between developers and would-be users, e.g., in the development of generic applications or mass-market software [35]. Furthermore, requirements identity risks relate to the availability of requirements; high risk means that the requirements are unknown or indistinguishable. Second, Mathiassen et al. accentuates **requirements volatility** as another key risk in software development [11, 36, 37]. Requirements volatility relates to the stability of requirements [11], [38]; high risk indicates that requirements change easily. Such dynamics occur when stakeholders learn more about the software during development or when internal or external conditions for using the software change. Finally, Mathiassen et al. [18] point out **requirements complexity** as a key risk in requirements and software development [39, 36]. Requirements complexity is a measure of how easy it is to understand requirements [38]; high risk, for instance, indicates that requirements are difficult to understand, specify, and communicate.

To conclude, we have shown in the previous section that the IS development practice has moved to more and more continuous ways of development. Furthermore, in this section, we have reviewed the requirements risk management literature, and we have shown that the current literature does not offer a continuous risk management or, more specifically, requirements risk profiling and methods. The literature does, however, offer foundations for developing such a method. We see that especially the contingency model by Mathiassen et al. [18] provides a theoretically firm ground for such a method development task. Similarly, we were inspired by the study by Keil et al. [24], which applied a Delphi survey to identify and rank the most important software development risks. In the following sections, we present first the chosen research methodology and then how we developed the proposed method.

#### 4. Research Methodology

We employed design science research [40-42] in this study as the research approach. Design science research complements both qualitative and quantitative research approaches by using the development and design of artifacts to assist in the formulation of theories and design of a novel and useful artifacts. According to Hevner et al. [40], the artifacts can be constructs, models, methods, and instantiations. Peffers et al. [19] extend this notion and reflect upon the thoughts of van Aken [43], adding that artifacts could also include social innovations or, as Järvinen [44] states, new properties of technical, social, and/or informational resources.

More specifically, we use the design science research methodology (DSRM) [19] to conduct our

research. The DSRM starts with the identification of research problem(s) and the motivation for the research. Based on evidence, reasoning, and inference, the process continues toward defining the objectives of a solution to solve the research problem. This process should be based upon prior knowledge in the given field of research. This knowledge is then used to design and develop an artifact, which is evaluated for its effectiveness or efficiency. This approach eventually leads to disciplinary knowledge.

For this study, we have applied the DSRM to structure and conduct our research. The presentation of the study has been informed by Gregor and Hevner [45]'s guidelines of how to present design science research. To develop the artifact, the *continuous requirements risk profiling method*, we used two research methods for data gathering as well as engaging and interacting with the industry participants. First, we applied the focus group interviewing method to identify requirements risks. Second, we followed Keil et al. [24] in using a Delphi survey to develop a project requirements risk profiling method, which is evaluated in the second Delphi round with the industry experts. For the evaluation of the method, we apply evaluation pattern approach as suggested by [46]. This process coincides with Peffers et al. [19] DSR process, but argues that evaluation should be done in a continuous manner during the research project. We provided details of the application of the focus group interviewing method and the Delphi survey in respective sections of the paper. This research process is depicted in Table 1.

**Table 1. Continuous artifact development and evaluation process in the study**

Delphi Iterations	Objective
Pre-Delphi Focus Groups	Identify risk themes and their underlying risk items
Delphi 1.1	Identify level of impact for each risk item, adding possible new items
Delphi 1.2	Assess the impact level of risk items
Delphi 1.3	Reach consensus on the level of impact of each risk item on ISD project
Delphi 2.1	Identify ISD phases most likely affected by each risk item
Delphi 2.2	Assess ISD phases most likely affected by each risk item
Delphi 2.3	Reach consensus on ISD phases most likely affected

#### 5. Pre-Delphi Focus Groups

We applied the focus group interviewing method as the data collection method in identifying requirements risk items and their fit to requirement risk themes. The strength of the focus group method is that it enables the focus group moderator (the researcher) to gather a homogeneous group of participants, who can easily

relate to each other due to their similar backgrounds and expertise in a particular field, to discuss a topic that is of interest to the organizational unit [47].

We invited participants who held the following titles: project manager, IT manager, application systems developer, product manager, business analyst, and consultant, among others. The number of focus groups to be conducted is often determined by the available budget [48]. We have chosen to include participants from three major New Zealand cities: Auckland, Wellington, and Christchurch. At the end of the recruitment process, we had 25 participants that were symmetrically distributed across three selected cities. Each focus group discussion lasted about 2 hours. To bring participants into discussion mode, the underlying objective of this research was outlined to participants at the beginning of the session: “What sources of risks there are involved in requirements development of the system development project?”

### 5.1. Thematic Data Analysis

Thematic analysis was employed as the data analysis method [49]. Thematic analysis is widely used by qualitative researchers to identify and analyze themes across qualitative data collected [49]. Thematic analysis focuses on producing and applying codes to data and is a special approach to dealing with theme development. A core step of thematic analysis is to perform a thematic coding exercise. In this study, open coding was used to assist the coding exercise. A code is a category that manifests meanings of a particular subset of data, and by grouping similar data under relevant codes, a researcher can further aggregate these codes into one main theme. In other words, a theme represents patterned meanings of the data, and it is strongly related with the research question [49]. Table 2 shows the six steps of thematic analysis [49] and the application of the steps in this study.

The coding software NVIVO 8 was used in generating the initial codes. The similar codes were grouped together to find candidate themes. Four candidate themes emerged from the analysis, whereas each candidate theme also contained many subthemes aggregated from texts. However, we discovered that some codes did not seem to fit into any candidate themes nor its subthemes; therefore, we temporarily placed them into a different theme called “miscellaneous” in order to review them later [49]. The purpose of reviewing candidate themes was to ensure that each candidate theme matched with meanings of its embedded codes and that all candidate themes could accurately depict a meaningful story of the data set [49]. We scanned through details of all candidate themes, and we concluded that these are the “real themes” and of significant value, as our data had shown that they can well represent the types of requirement risks in ISD projects. We also assessed each theme to ensure that there is no overlap, so each theme could be clearly

distinguished from those of others. A common guideline in revising themes is that by looking at the heading of each theme, the researcher can clearly describe its meanings with fewer than 2 sentences [49]. Table 3 presents the found risk items.

**Table 2. Thematic analysis process [49]**

Name of Step	Implementation of the Step in this Study
Familiar with data set	Transcribe discussions. Read through all focus groups' data.
Generating initial codes	Identify interesting features and encapsulate meanings with codes. Assimilate similar data into relevant codes respectively.
Searching for themes	Group similar codes together. Use themes to represent shared meanings.
Reviewing themes	Produce a thematic map of the analysis (incl. checking representativeness and consistencies of generated themes).
Defining and naming themes	Revise and rename themes as necessary to avoid ambiguity and misunderstanding.
Producing analysis report	Produce reports showing the results, along with justifications on how these results relate back to the research objective and literature.

We found that the themes “identity,” “volatility,” and “complexity” are consistent with Mathiassen [18]’s risk dimensions, respectively. The way in which Mathiassen et al. [18] synthesized requirements development risks into three easily understood categories is similar with the emerging themes from our results. In other words, with the inputs received from industrial practitioners, we found a positive match between the literature and practice regarding the dimensionalities of requirements development risks. Our results demonstrate a positive confirmation of the relative importance of Mathiassen et al. [18]’s risk dimensions. An example extract concerning an “ambiguous requirements risk item” follows:

*“One of the risks in requirements is, they [are] vague. They are not very well defined. They can be interpreted in multiple manners. That’s one of the risks in requirement gathering. We understood like this, and you understood like that, and now we are in trouble. It’s the vague requirements that [are] causing the major risk” - Kevin, Senior Project Manager*

However, we did also recognize that there were a number of requirements risks that did not fit into these three themes [18]. We labeled these as integrity risks. This finding is beyond the scope of this paper, but we will return this matter and its possible implications for research in the discussion section of the paper.

**Table 3. Requirement risks identified in focus groups**

Absence of project sponsor
Access to clients
Ambiguous requirements
Change in business strategy and direction
Change in external regulation
Choice of development methodology
Client commitment
Communication medium
Compliance with external regulations
Conflicting requirements
Constrained by users' knowledge
Delivering what the client needs
Emerging requirements dependency
Emerging requirements dependency
Fixed budget and deadline
Hostile users
Incorrect stakeholders
Knowledge gap between coworkers
Lack of collaboration
Lack of requirement review
Misunderstood business needs
Negative perception
Omitted requirements
Organizational politics
Original source of requirements (pre-elicited versus newly captured)
Project team member turnover
Technology changes
Top management support
Translation from conceptual requirements to detailed specification
Underestimation of the magnitude of changes
Unrated requirements
Verbal precision

## 6. Delphi 1

The Delphi method has been used in information systems research previously by [23-24, 50], for example. This method lends itself very well in areas where prior research is limited and in research that is exploratory in nature. Delphi study involves gathering opinions from a panel of experts in a particular field and trying to achieve a level of consensus from the panel [51]. Information supplied by the panel members is shared among the panel members. However, the individual panel members themselves are not identified when the information is shared. This provides a medium for panel members to voice their opinions and reasoning safely without being labeled or becoming the target of any kind of peer pressure.

For the purposes of this study, 15 was an ideal number of panel members, given the experience that they brought to the panel. The members had a minimum of 5 years of experience and have performed one of the

following roles: project manager, business analyst, or technical architect.

Participants who formed the panel were chosen based on the selection criteria and based on the first 15 who showed interest in the study. On average, the work experience in the panel was 16 years. Panel members had undertaken a variety of roles over the years in their industry, ranging from programming roles to roles as development managers, IT managers, delivery managers, integration consultants, team leaders, etc.

The Delphi study was carried out in two main rounds (Table 1). Each round included three iterations. The focus of Round 1 was to understand the level of impact each individual risk item can cause on an ISD project. The focus of Round 2 was to understand which phase was most likely to be impacted by the risk item and any other phases that are also likely to be impacted. Most Delphi studies start with an explicit brainstorming round. In this study, the risk item used for the purposes of gaining an agreement from the panel members were sourced from the conducted focus group interviews, and the risk items identified were developed as a result of a focus group study. Generally, Delphi studies aim to get a level of agreement among panel members on the topic or issue that is being discussed or explored. In this study, free-marginal kappa was used to determine the level of agreement among panel members. Free-Marginal Kappa was selected because the panel members were to select a particular proportion response, be it risk impact level or the phase that can be affected by a particular risk item [52].

The goal of Delphi Round 1 was to identify the level of impact of each risk item on the ISD project. Round 1 was divided into three iterations.

**Iteration 1.** Delphi study's Round 1 started with 32 risk items that were identified as a result of the focus group study. These 32 risk items are provided in Table 3. Each risk was provided with a brief description and included in the survey form to the panel, which asked each panel member to indicate the level of risk. The levels of impact that were of interest were high, medium, and low. Panel members were encouraged to provide feedback regarding the impact level for various risk items so as to provide more insight into the reason that certain risks were considered to have a high, medium, or low impact. Panel members were also provided with the opportunity to add additional risk items during the first iteration if they felt that certain requirements risks were being left out of the study.

Panel members were allowed 2 weeks to return the survey form. Once the survey form was returned, the overall results showed that there was only a slight agreement among panel members regarding the impact that various risk items can have on ISD projects (free-marginal kappa = 0.1699). There were 14 risk items that the majority of the panel members had agreed on with regards to the impact that these risk items can have on ISD projects. However, the feedback that we received from the panel also indicated that the impact on ISD

projects for each risk item was subject to other factors, such as type of project, duration of project, etc. The responses also added 5 risk items to the risk items (Table 4).

**Table 4. Risk items added by panel members during Iteration 1 – Round 1**

Risk	Description Provided By Panel Members
Key team members not compensated adequately	Key team members can be pressured to work additional hours to meet timeframe expectations.
Team conflict	Conflict develops among team members who need to work together.
Quality of testing	Time or other pressures create gaps between the testing time needed or planned and the time spent testing.
Gap between actual project status and status reported	This gap normally arises by those who report a different status from the actual one due to their belief that time will be made up and the gap will go unnoticed.
Gap between skills needed and resourced	There is a poor fit between skills needed and those recruited.

**Iteration 2.** The responses from Iteration 1 were collated and supplied back to the panel along with the feedback received from the panel for Iteration 2. The survey in Iteration 2 also included the additional risk items. Panel members were informed that they could change the impact level they had assigned to a particular risk item if they chose to agree with the majority of the panel members if they had not done so as part of the first iteration or if they wanted to change the impact level based on the feedback received. Panel members were again encouraged to provide additional feedback.

The responses from Iteration 2 showed that some panel members had chosen to change their impact perceptions for various risk items based on the perception of the majority of the members in the panel along with the feedback that was received from Iteration 1. The responses from Iteration 2 showed an improvement in the level of agreement (free-marginal kappa = 0.248869). It started to be clear that the impact that different risks can cause could vary depending on the project and the overall experience that a panel member had in dealing with some of the risk items. The additional risk items that were added also did not seem to get a clear indication from all panel members regarding whether they agreed that the new risks were, indeed, requirements-related risk items (free-marginal kappa = 0.139048) or what type of requirements risks they were if they indicated that they considered the risk

to be a requirements-related risk. For this reason, the additional risk items were not included in Round 2.

**Iteration 3.** A summary survey was initiated to determine whether the panel members agreed with the view that was perceived to be the message of the panel. The results showed that in treating both “strongly agree” and “agree” as separate responses, there was fair agreement (free-marginal kappa = 0.343915) among the panel members. But when treating the responses from panel members who indicated that they agreed and panel members who indicated that they strongly agreed to mean that the member actually agreed with the perception, there was near-perfect agreement among the panel members (free-marginal kappa = 0.86665).

The responses also showed that some of panel members had chosen to change the impact perceptions of some of the risk items, resulting in 6 additional risk items now showing that a majority of panel members agreed on their impact perceptions (1 risk item perceived as causing a high impact, 5 risk items causing a medium impact). These additional risk items were included in Round 2 as a carryover from Round 1. Panel members were asked whether they agreed on the perceptions of the impact in a similar manner to how they were asked in the summary round in Round 1. The responses for this indicated that the panel agreed on the impact perception (free-marginal kappa = 0.714285).

The risk items identified in Table 5 provide the list of risk items and their impact perceptions that the panel members agreed upon.

## 7. Delphi 2

The goal of Delphi Round 2 was to connect risk items to ISD phases. The main aim of Round 2 was to identify the phase that was most likely to be affected by a risk item. For the purposes of this study, three main phases of a continuous IS development project were considered: requirements, design, and implementation phases. Only the risk items for which the panel reached an agreement regarding the impact to an ISD project were included.

**Iteration 1.** In Iteration 1, the panel was asked to choose the phase most likely to be affected by each risk item. They were also asked to select other phases that were also likely to be affected by the risk item. Panel members were allowed two weeks to return the survey form, and they were instructed to choose only one phase as the phase “most likely” to be affected by a risk item. They were also instructed that they were able to choose more than one phase by indicating which other phases were “also likely” to be affected. They were asked not to select the phase that they had selected as the phase most likely to be affected when selecting the phases that were also likely to be affected. Once the survey form was returned, the overall results showed that there was a fair agreement among the panel members regarding the

**Table 5. Impact perception agreed on by the panel**

Impact to ISD Project	Risk Items
High	<ol style="list-style-type: none"> <li>1. Absence of project sponsor</li> <li>2. Access to clients</li> <li>3. Ambiguous requirements</li> <li>4. Client commitment</li> <li>5. Delivering what the client needs</li> <li>6. Incorrect stakeholder</li> <li>7. Missing requirements</li> <li>8. Misunderstood business needs</li> <li>9. Unrated requirements</li> </ol>
Medium	<ol style="list-style-type: none"> <li>10. Change in business strategy and direction</li> <li>11. Change in external regulations</li> <li>12. Compliance with external regulations</li> <li>13. Conflicting requirements</li> <li>14. Choice of development methodology</li> <li>15. Emerging requirements dependency</li> <li>16. Fixed budget and deadline</li> <li>17. Hostile users</li> <li>18. Knowledge gap between coworkers</li> <li>19. Lack of collaboration</li> <li>20. Project team member turnover</li> <li>21. Underestimation of change magnitude</li> </ol>
Low	<ol style="list-style-type: none"> <li>22. Constrained by users' knowledge</li> <li>23. Technology changes</li> </ol>

phase that was most likely to be affected by each of the 23 risk items (free-marginal kappa = 0.21145). There were only 6 risk items for which the majority of the panel members had indicated that the same phase was the most likely to be affected. There was only slight agreement on the responses regarding which other phases were also likely to be affected by the risk items (free-marginal kappa = 0.0756491). For the responses for “also likely to affect phases,” participants seemed to be divided on which other phases were also likely to be affected. Some participants seemed to think that risk is only likely to affect one phase, while others thought that it could affect all phases. The results were collated along with the panel feedback received, and Iteration 2 was initiated.

**Iteration 2.** In Iteration 2, the collected data from Iteration 1 was evaluated by the panel members. The responses showed that some panel members had chosen to change their responses for the phase most likely to be affected based on the perception of the majority of the members in the panel along with the feedback that was received from Round 2 Iteration 1. The responses from Iteration 2 showed an improvement in the level of agreement for the phase most likely to be affected (free-

marginal kappa = 0.33043). However, the same could not be said for panel responses that indicated the other phases that were also likely to be affected. This still showed only a slight agreement with a slight improvement on the kappa values (free-marginal kappa = 0.16352).

At this point, it started becoming clear that it is unlikely to get an agreement regarding the phases of an ISD project that could also be affected by the risk items. The panel feedback seemed to point out that if a risk item was not addressed at a particular phase, any other phase of the project could be affected. Because of this, it was decided to eliminate the question of which other phases were also likely to be affected by a risk item. It was replaced with a statement that “a risk item that has not been identified, or had not been properly addressed in a timely fashion, can affect any phase of the project.” The panel members were asked whether they agreed with that statement.

**Iteration 3.** A summary survey was initiated to determine if panel members agreed with the final view of the panel members. The results of the survey showed that regarding treating both “strongly agree” and “agree” as separate responses, there was moderate agreement (free-marginal kappa = 0.475132) among panel members about the choice requirements risks that affected the requirements phase, design phase, and implementation phase of an ISD project. But when treating the responses from panel members who indicated that they agreed and panel members who indicated that they strongly agreed to mean that the member actually agreed with the perception, there was near-perfect agreement among the panel members (free-marginal kappa = 0.999999).

The risk items identified in Table 6 provide the risk items and the corresponding phase that they are most likely to affect. The results from the summary iteration also suggest that even though certain risk items have been perceived and agreed upon by panel members as most likely to affect a particular phase of ISD projects, the risks can affect any phase of an ISD project if they are not managed properly. The panel members’ views when considering “strongly agree” and “agree” as separate responses showed fair agreement (free-marginal kappa = 0.288888). The panel members’ view on this when considering both “strongly agree” and “agree” to mean that the panel agreed on that view showed near-perfect agreement on the view (free-marginal kappa = 0.999999).

For the remaining 6 risk items, the panel members were not able to agree on the phases that were most likely to be affected. However, for these 6 risk items listed in Table 6, based on the panel’s agreement with the statement that “a risk item that has not been identified, or had not been properly addressed in a timely fashion, can affect any phase of the project” these 6 risks could be considered as risk items that have the potential to affect any phase.

**Table 6. Project risk profiling method**

Requirements Phase		Design Phase		Implementation Phase		
Risk	Impact	Risk	Impact	Risk	Impact	
Absence of project sponsor	High	Missing requirements	High	Hostile Users	Medium	
Access to clients (proximity to source)	High	Delivering what the client requires	High	Project team member turnover	Medium	
Incorrect stakeholder	High	Compliance with external regulations	Medium			
Misunderstood business needs	High	Conflicting requirements	Medium			
Change in business strategy and direction	Medium	Development methodology choice	Medium			
Constrained by users' knowledge	Low	Emerging requirements dependency	Medium	<b>Risks Affecting All Phases</b>		
		Knowledge gap between coworkers	Medium			
		Lack of collaboration	Medium			
		Technology changes	Low			
Ambiguous requirements	High	Ambiguous requirements	High	Ambiguous requirements	High	
Unrated requirements	High	Unrated requirements	High	Unrated requirements	High	
Client commitment	High	Client commitment	High	Client commitment	High	
Change in external regulations	Medium	Change in external regulations	Medium	Change in external regulations	Medium	
Underestimation of change magnitude	Medium	Underestimation of change magnitude	Medium	Underestimation of change magnitude	Medium	
Fixed budget and timelines	Medium	Fixed budget and timelines	Medium	Fixed budget and timelines	Medium	

## 8. Discussion

Our design science research study has developed a continuous requirements risk profiling method. The developed method can also be considered as a nascent DSR theory that summarizes knowledge as operational principles or architecture [46]. Our profiling method is summarized in Table 6. The method is founded on the continuous development framework (Figure 1) and research on contingency theory-based requirements risk management [18]. The method provides three elements to accomplish continuous risk profiling for requirements risks.

Our method provides analysts with the tools to characterize the risk profile of a project dynamically. In other words, by using our profiling method, the analyst is able to accomplish the following: 1) identify the key risks for the project, 2) assess the current risk profile of the project, and 3) propose how to resolve the risks in a prioritized manner by using the indicated potential impact of a risk to a project. The method is also considered to be agnostic of the underlying IS development methodology and can be used with traditional stage-based or agile ISD methodologies.

We also recognize that the current work does not yet fully address the need for constant profiling and risk

management. First of all, we see that we should more carefully focus on how to do continuous risk management. This would mean that we would need to re-assess how the current contingency-based risk management models and frameworks, such as [18], address the needs of continuous risk management. So far, this has been addressed through stage-based development approaches; see, e.g., [10-13].

We also argue that a more detailed method is needed to address how to assist the analysts to do constant profiling. We assume that there will be iteration cycles between three key elements of a method: identify risks, assess overall risk profile, and intervene. We are currently working on developing tools, e.g., for more detailed risk identification, which we deem important.

Furthermore, Delphi method is in its nature subjective and relies the experiences of the participants. This brings forward an interesting of objectivity in risk measurement. The current literature does well not address how certain problems occur in a project and of the consequences of those problems if they occur. The contingency risk management literature [10-13, 15, 18] does not really address the issue of what are the objective measures of probability for such project events. Moreover, the literature provides different

contingency rules of how to address certain project related risks, see, e.g. [18].

In addition, in our focus group phase of the study, we found that some of the requirements risks do not fit into the three earlier suggested risk categories, i.e., identity, volatility, and complexity. The previous contingency models in the literature, such as [11, 18], do not discuss “integrity”-type requirements risks. We believe that this new integrity risk category should be added into the contingency models to reflect an updated understanding of requirements risks.

Finally, our paper does not address how to intervene and resolve the requirements risks. For this, we should link the profiling method to the use of requirements resolution techniques. Mathiassen et al. [18] have proposed the use of specific resolution technique types for different requirements risks. Our current work does yet provide means for this end. Therefore, we see that the presented risk profiling method should be extended and linked more directly to the contingency-based risk management literature. We see certain work [18] especially interesting for this purpose.

## 9. Conclusions

This paper proposes a continuous requirements risk management method that is founded on the previous literature contingency-based risk management and how to identify and mitigate requirements risks. We developed the method in cooperation with industry professionals in New Zealand. The study applies design science research methodology [19] to structure and guide the research process as well as focus group interviews and a Delphi study for data collection. We argue that the developed method can be integrated into ISD projects using different ISD methods, such as structured or agile or combinations of both.

Our paper contributes to the literature by presenting a design science research study, which has rigorously applied one of the prominent DSR methodologies available today. The study provides an exemplar of how to develop a nascent DSR theory—in this case, a requirements risk profiling method. The study also demonstrates how different data collection methods can be used for method development when applying design science research methodology [19]. The developed method contributes to the literature by providing an up-to-date requirements risk item list and a continuous requirements risk profiling method.

Although we have strived to conduct a rigorous and well-planned DSR study, we do recognize some limitations. First of all, the number of participants in the Delphi study could have been bigger in order to improve the generalizability of the results. However, we were satisfied that the current number of participants ( $n=15$ ) was in line with the previous literature that has applied the Delphi method in IS research [24, 50]. Thus, we do not consider this as a major limitation for the

study. However, in future research, it would be interesting to conduct several Delphi studies in parallel on a singular research topic to see how this would enhance or change the results. This would, however, require substantial resources to run the study, as conducting Delphi studies is a laborious activity.

## References

- [1] Royce, W.W., "Managing the Development of Large Software Systems: Concepts and Techniques," Proceedings of the 9th International Conference on Software Engineering, Monterey, CA , pp. 328 – 338, 1970
- [2] Barlow, J.B., J.S. Giboney,, M.J. Keith, D.W. Wilson, R.M. Schuetzler, P.B. Lowry, A.Vance, "Overview and Guidance on Agile Development in Large Organizations," Communications of the Association for Information Systems: Vol. 29, Article 2, 2011.
- [3] Avison, D., G. Fitzgerald, G., Information Systems Development, methodologies, techniques & tools. Fourth Edition. London, McGraw-Hill, 2006.
- [4] Williams, L., "What Agile Teams Think of Agile Principles," Communications of the ACM, Vol. 55, No. 4, Pages 71-76, 2012.
- [5] Baskerville, R.L., J. Pries-Heje,, "Short Cycle Time Systems Development," Information Systems Journal 14(3), pp. 237-264, 2004.
- [6] Fowler, Continuous Integration. <http://martinfowler.com/articles/continuousIntegration.html>, accessed 9th September 2013, 2006.
- [7] Ståhl, D., J. Bosch , J., "Modeling Continuous Integration Practice Differences in Industry Software Development," Journal of Systems and Software, In Press.
- [8] Cao, L., B. Ramesh, "Agile Software Development: Ad Hoc Practices or Sound Principles?" IT Pro, March / April, p. 41-47, 2007.
- [9] Muthitacharoen, A., K.A. Saeed K.A., "Examining User Involvement in Continuous Software Development (A case of error reporting system)," Communications of the ACM Sep2009, Vol. 52 Issue 9, p113-117, 2009.
- [10] Alter, S., M. Ginzberg, "Managing uncertainty in MIS implementation," Sloan Management Review (20:1), pp 23-31, 1978.
- [11] Davis, G.B., "Strategies for information requirements determination" IBM Systems Journal (21) 1, pp. 4-30, 1982.
- [12] McFarlan, F.W., "Portfolio Approach to Information Systems," Harvard Business Review (59:5), pp 142-150, 1981.
- [13] McFarlan, F.W., "Portfolio Approach to Information Systems," Journal of Systems Management (33:1), pp 12-19, 1982.
- [14] Barki, H., S. Rivard, J. Talbot, "Toward an Assessment of Software Development Risk," Journal of Management Information Systems (10) 2, pp. 203-225, 1993.
- [15] Saarinen, T., A. Vepsäläinen, "Managing the risks of information systems implementation," European Journal of Information Systems (2:4), pp 283-295, 1993.
- [16] Lytynen, K., L. Mathiassen, J. Ropponen, "Attention shaping and software risk - A categorical analysis of four classical risk management approaches," Information Systems Research (9:3) Sep, pp 233-255, 1998.

- [17] Hickey, A. M., A. Davis, "A Unified Model of Requirements Elicitation," *Journal of Management Information Systems* (20:4), pp 65–84, 2004.
- [18] Mathiassen, L., T. Saarinen, T. Tuunanen, M. Rossi, M. "A Contingency Model for Requirements Development," *Journal of Association of Information Systems* (8:11) November, pp 569-597, 2007.
- [19] Peffers, K., T. Tuunanen, M. Rothenberger, S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems* (24:3), pp 45-77, 2008.
- [20] Truex, D.P., R. Baskerville, H. Klein, "Growing Systems in Emergent Organizations," *Communications of the ACM*, Vol. 42, No. 8. 117-123, 1999.
- [21] Martin, R.H., D. Raffo, "A Model of the Software Development Process Using Both Continuous and Discrete Models," *Software Process Improvement and Practice*, 5, pp. 147-157, 2000.
- [22] Boehm, B.W., "Software Engineering," *IEEE Transactions on Computers* (C-25) 12, pp. 1226-1241, 1976.
- [23] Schmidt, R., K. Lyytinen, M. Keil, P. Cule, "Identifying Software Project Risks: An international Delphi Study," *Journal of Management Information Systems* (17) 4, pp. 5-36, 2001.
- [24] Keil, M., P.E. Cule, K. Lyytinen, R.C. Schmidt, "A Framework for Identifying Software Project Risks," *Communications of the ACM* (41) 11, pp. 76-83, 1998.
- [25] Boehm, B.W. "A Spiral model of software development and enhancement," *Computer* (21) 5, pp. 61-72, 1988.
- [26] Beck, K., "Embracing change with eXtreme programming," *IEEE Computer* (32) 10, pp. 70-77, 1999.
- [27] Davis, A. M., E. H. Bersoff, E.R. Comer, "A Strategy For Comparing Alternative Software Development Life Cycle Models," *IEEE Transactions on Software Engineering* (14) 10, pp. 1453-1461, 1988
- [28] Luqi, "Software Evolution Through Rapid Prototyping," *Computer* (20) 5, pp. 13-25, 1989.
- [29] Schwaber, K., "SCRUM Development Process," Workshop Report: Business Object Design and Implementation. 10th Annual Conference on Object-Oriented Programming Systems, Languages, and Applications. Addendum to the Proceedings, 1995.
- [30] Conrow, E.H., P.S. Shishido, "Implementing Risk Management on Software Intensive Projects," *IEEE Software* (14) 3, pp.83-89, 1997.
- [31] Boehm, B. W., "Software Risk Management: Principles and Practices," *IEEE Software* (8) 1, pp. 32-41, 1991.
- [32] Tiwana, A., M. Keil, "The One Minute Risk Assessment Tool," *Communications of the ACM* (47) 11, pp. 73 -77, 2004.
- [33] Wallace, L., "The development of an instrument to measure software project risk", unpublished Doctoral Dissertation, Georgia State University, 1999.
- [34] Wallace, L., M. Keil, R. Arun, "Understanding software project risk; a cluster analysis," *Information & Management* (42) 1, pp. 115-125, 2004.
- [35] Regnell, B., M. Höst, J. Natt och Dag, P. Beremark, "An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software," *Requirements Engineering Journal* (6) 1, pp. 51-62, 2001.
- [36] Fazlollahi, B., M. R. Tanniru, "Selecting a Requirement Determination Methodology - Contingency Approach Revisited," *Information & Management*, Vol.21, No.5, pp. 291-303, 1991.
- [37] Lyytinen, K. "Different Perspectives on Information-Systems - Problems and Solutions," *ACM Computing Surveys* (19) 1, pp. 5-46, 1987.
- [38] Mathiassen, L., T. Seewaldt, J. Stage, "Prototyping and Specifying: Principles and Practices of a Mixed Approach," *Scandinavian Journal of Information Systems* (7) 1, pp. 55-72, 1995.
- [39] Brooks, F. P., "No Silver Bullet - Essence and Accidents of Software Engineering," *IEEE Computer* (20) 4, pp. 10-19, 1987.
- [40] Hevner, A.R., S.T. March, J. Park, "Design Research in Information Systems Research," *MIS Quarterly* (28:1), pp 75-105, 2004.
- [41] March, S., G. Smith, "Design and Natural Science Research on Information Technology," *Decision Support Systems* (15), pp 251-266, 1995.
- [42] Walls, J., G. Widmeyer, O. El Sawy, "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research* (3:1), pp 36-59, 1992.
- [43] Van Aken, J.E., "Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules," *Journal of Management Studies*, 41, 219-246, 2004.
- [44] Järvinen, P., "Action Research is Similar to Design Science," *Quality & Quantity* (41:1) February, pp 37-54, 2007.
- [45] Gregor, S., A.R., Hevner, "Positioning and Presenting Design Science Research for Maximum Impact," *MIS Quarterly* (37:2), 2013.
- [46] Sonnenberg, C., J. vom Brocke, "Evaluation in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research" K. Peffers, M. Rothenberger and B. Kuechler: *Proceedings of the Sixth International Conference on Design Science Research in Information Systems and Technology 2012*, Lecture Notes on Computer Science 7282, Springer, pp. 381-397.
- [47] Morgan, D.L., *Focus Groups as Qualitative Research* (2nd), California: Sage Publications, 1997.
- [48] Greenbaum, T.L., *The Handbook for focus Group research* (2nd), California, Sage Publications, 1998
- [49] Braun, V., V. Clarke, V., "Using thematic analysis in psychology," *Qualitative research in psychology*, Vol.3, pp.77-101, 2006.
- [50] Akkermans, H.A., P Bogerd, E Yücesan, L.N. van Wassenhove, "The impact of ERP on supply chain management: Exploratory findings from a European Delphi study," *European Journal Of Operational Research*, 146, pp. 284-301, 2003.
- [51] Linstone, H.A., M. Turoff, "The Delphi Method: Techniques and Application" Retrieved 20/03/2009 from, <http://www.is.njit.edu/pubs/delphibook/>, 2002.
- [52] Brennan, R.L., D.J. Prediger, "Coefficient Kappa: Some uses, misuses, and alternatives," *Educational and Psychological Measurement* (41) 3, pp. 687-699, 1981.