

Spring 3-23-2010

UNDERSTANDING THE EMERGENCE OF REQUIREMENT RISKS IN INFORMATION SYSTEMS PROJECTS

Nipon Parinyavuttichai
The University of Sheffield, nparinya@gmail.com

Angela Lin
University of Sheffield, A.Lin@Shef.ac.uk

Follow this and additional works at: <http://aisel.aisnet.org/ukais2010>

Recommended Citation

Parinyavuttichai, Nipon and Lin, Angela, "UNDERSTANDING THE EMERGENCE OF REQUIREMENT RISKS IN INFORMATION SYSTEMS PROJECTS" (2010). *UK Academy for Information Systems Conference Proceedings 2010*. 43.
<http://aisel.aisnet.org/ukais2010/43>

This material is brought to you by the UK Academy for Information Systems at AIS Electronic Library (AISeL). It has been accepted for inclusion in UK Academy for Information Systems Conference Proceedings 2010 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

UNDERSTANDING THE EMERGENCE OF REQUIREMENT RISKS IN INFORMATION SYSTEMS PROJECTS

Nipon Parinyavuttichai

The University of Sheffield, Sheffield UK
Email: N.Parinyavuttichai@Shef.ac.uk

Angela Lin

The University of Sheffield, Sheffield UK
Email: A.Lin@Shef.ac.uk

Abstract

Information Systems (IS) requirement risks are one of the most important sources that contribute to project problems, such as escalation of project cost and schedule. Early identifying and managing requirement risks is therefore an important task to avoid project complexity and increase chance of project success. Earlier research has identified various requirement risks in IS project. However, there has been little research on explaining the emergence of requirement risks. This paper proposes some initial insights into the origins of requirement risks based on a case study of an IS project having requirement risks. The results of this study suggest that the emergence of requirement risks can be identified and explained from various IS development (ISD) practices and some organisational behaviours perspectives. Moreover, requirements risk can occur not only in the requirement collection and analysis phase, but also in the later phases of the ISD. Conclusion and implications for future research are also provided.

Keywords: requirement risk, socio-organisational perspective, requirement determination, IS failure, Emergence, information systems development, Escalation

1.0 Introduction

How to effectively manage information systems projects in order to deliver systems on time and within budget has been a long standing topic in the information systems field (Keil and Mann, 1997; Barki et al., 2001; Pan et al., 2004). Failures in IS project can be seen in various forms including project delay in systems delivery (Barki et al., 2001), cost overrun (Keil et al., 1998), project abandonment (Boehm, 2000), etc. Numerous factors that contribute to project management failures have been identified such as requirement risks (Schware and Bhatnagar, 2001), inappropriate project management (Kim et al, 2005), and escalation behaviours (Keil and Mann, 1997). Among these contributing factors requirement risks may be mentioned frequently but they are least discussed in the IS literature (McEwen, 2004, Verner et al., 2005). For example, it is reported that nearly 60% of defects in information systems come from

poor requirement management (Williams and Kennedy, 1999, McEwen, 2004); and problems associated with requirements are accountable for nearly half of the problems encountered in most IS projects while only 16% of the IS projects were successfully developed without any requirements problems (Glass, 2001, Hall et al., 2002). It is also evident that poor IS requirements has adverse effects on IS projects and is the primary source of subsequent project complexities and risks, e.g. escalation of project cost and schedule (Kasser, 2002; Ayoo and Lubega, 2009). Identifying risks associated with project requirements in the early stage of an information systems development and managing them properly is therefore an important task as subsequent project complications that are induced by requirements risks can be avoided or mitigated (Shull et al., 2000; Han and Huang, 2007).

Many studies of requirement risks in ISD focus attention on identifying types and causes of requirement risks (Weigers, 2000; Verner et al., 2005; Ayoo and Lubega, 2009). The findings of these studies although contribute to the general understanding of what types of requirement risks can occur, they do not explain well the effects of requirement risks on the subsequent project risks and why requirement risks can be observed in any stage of an IS project. This paper hence aims to address this gap in the current literature by examining the reasons for the emergence of requirement risks and whether and how they induce other subsequent risks.

The organization of the paper is as follows. The next section is theoretical background which is the literature review of requirement risks in ISD. Section three describes research strategy adopted by the study and the research site where the fieldwork was taken place. Section four presents the findings of the case study and this is followed by the discussion section. Section six concludes the paper and provides implications of the study for practice and further research.

2.0 Theoretical Background

Oberg et al., (2000) defines software requirements as “a condition or capability to which the system being built must conform.” Software requirements contain business objectives and activities that aim to enhance user organisation’s practices (Hickey and Davis, 2004; Bleistein et al., 2006). They consist of characteristics of system features, users’ views on the existing system, requests for future system, etc (Hickey and Davis, 2004; Bleistein et al., 2006). Requirement risks are generally referred to uncertainties that arise from the differences between the actual requirements of users for a system and the requirements perceived by system developers (Daft and

Macintosh, 1981). Those uncertainties will affect decisions made about systems design and ultimately create subsequent project risks (Nidumolu, 1996). Table 1 summarises some requirement risks identified from the literature and among them incomplete (or inadequate) requirements, changing requirements, and misunderstanding (or incorrect) requirements are the three types of requirement risks that have been identified by various studies to be important.

Requirements Risk	Definition	Reference
Inadequate or incomplete requirements	User specifications are overlooked.	Addison (2003), Ayoo and Lubega (2009), Borland Software Corporation (2005), Dey et al. (2007), Gottesdiener (2009), Howcroft and Wilson (2008), Kumar (2002), Laussen and Vinter (2001), Boehm (2000)
Changing requirements	Happen in the situation where system functionalities or user requirements keep changing in IS projects.	Ayoo and Lubega (2009), Carter et al. (2001), Demarco and Lister (2003), Dey et al. (2007), Fowler (2001), Gottesdiener (2009), Laussen and Vinter (2001), Pare et al. (2008), Sumner (2000), Tiwana and Keil (2004), Boehm (2000)
Misunderstanding or incorrect requirements	Uncertainty arisen from the situation where users and developers have different views of the system	Addison (2003), Dey et al. (2007), Gottesdiener (2009), Laussen and Vinter (2001), McAllister (2006), Pan et al., (2004), Sumner (2000), Wiegers (2000)
Voluminous requirements	Excessive amount of requirements that causes disagreements among project stakeholders due to a variety of interpretation of the technical terms used in requirements collection process.	Robinson et al. (2003)
Complicated requirements	Occur in the situation in which project stakeholders intend to ignore the development of some requirements in order to avoid conflicts among the stakeholders. Complicated requirements also refer to the requirements that are too difficult to design.	Robinson et al. (2003)
Advanced requirements	The unfamiliar or new requirements for the project developers and users.	Schmidt et al., (2001)
Lack of prioritised requirements	Lack of clearly defining priority to the design of project requirements from the outset.	Wiegers (2000)

Table 1: Requirement risks in IS research

2.1 Incomplete Requirements Risk

Risk of incomplete requirements is listed as the top project risk in the IS literature and trade reports (The Standish Group, 1995; Kumar, 2002; Addison, 2003). Risk of incomplete requirements usually emerge when users are unable to articulate their requirements, project specifications are overlooked, requirements are not well documented, or requirements are ignored by developers during the requirements collection (Lauesen and Vinter, 2001). It is also found that lack of adequate user involvement in IS project contributes to the emergence of the incomplete requirements risks because developers may develop an information system based on their assumptions which are not consistent with the actual user requirements (Howcroft and Wilson, 2003).

Risk of incomplete requirement can lead to waste of resources. A project team may have to spend additional time, efforts, and costs to develop system functionalities that are required by the users but not included in the original design, or to fix the errors in design because of incomplete requirements (Lauesen and Vinter, 2001; McAllister, 2006; Gottesdiener, 2009). Williams and Kennedy (1999) report that a project team could spend up to 80% of the team efforts to fix the problems arising from the incomplete requirements risks. McConnell (1996) also observes that correcting downstream requirements errors could cost up to 50 to 200 times more than the cost of correcting the problems upstream.

2.2 Changing Requirements Risk

The second type of requirement risk is risk of changing requirements. Risk of changing requirements refers to the situations where system functionalities or user requirements change continuously throughout an IS project (Carter et al., 2001; DeMarco and Lister, 2003). A number of reasons have been suggested by the literature that contributes to changing requirements. Carter et al. (2001) notices that using evolutionary ISD approach, which allows changes of requirements throughout ISD, creates uncertainty in requirements. Changes in business environment where an information system will be implemented lead to changes in systems requirements. For instance, Jones (1996) finds that IS requirements are likely to change when there are changes in business strategy or impacts from external environments, e.g. changes in government policy or regulation. Teger (1980) also finds that project scope can change because of the conflict between business units in user organisations. This is

because each unit may have different project expectations for the project (Schmidt et al., 2001).

Sometimes changes in systems requirements are inevitable but it is not always easy to foresee the possible changes at the outset of the project and therefore the project team may feel less prepared when the changes actually happen (Fowler, 2001). By allowing change in requirements to happen, it is difficult to avoid escalation of commitment on a failing course of action (DeMarco and Lister, 2003; Tiwana and Keil, 2004; Pare et al., 2008).

2.3 Misunderstanding Requirements Risk

The third type of requirements risk is the risk of misunderstanding requirements. Misunderstanding requirements is the situation where both users and developers have different views of the system but they fail to communicate with each other (Addison, 2003). Misunderstanding requirements can lead to inaccurate or incorrect systems specifications (Wieggers, 2000; Dey et al., 2007; Gottesdiener, 2009). Misunderstanding requirements may arise if users and developers do not share the same interests (Sumner, 2000; Wieggers, 2000). For example users may pay more attentions to business aspects of a system while developers focus their attentions to technical aspects of the system (Wieggers, 2000).

Misunderstanding requirements can also occur because of physical distance and lack of communication between users and developers (Pan et al., 2004). It is found when users and developers are not working in close physical proximity and do not communicate frequently developers are likely to develop a system on the basis of their own understandings of the requirements and problems (Pan et al., 2004).

Risk of misunderstanding requirements often leads to consequences where project team needs to correct mistakes in systems specifications later in the project, revise systems design, change functionalities in the system, and change project documentation (Wieggers, 2003; McAllister, 2006; Gottesdiener, 2009).

3.0 Research Methodology

3.1 Research Strategy

This paper is based on a research project carried out to understand risk management in information systems development projects in Thailand. The research employed a qualitative case study approach. The main reasons for choosing this approach are first,

with the approach we are able explore possible causes of requirement risks with an open mind (Hodkinson and Hodkinson, 2001). This is important as the research takes the stance that each IS project is unique and therefore risks that each project faces should be investigated in its context. Second, the approach allows us to investigate how requirement risks emerged in an IS project from the perspectives of project team members involved (Stake, 1995; Creswell, 2007; Creswell, 2009; Yin, 2009). Third, with qualitative case study approach rich data from multiple sources can be collected systematically in order to help us have insights into the case. The multiple sources of data can also be used to enhance credibility of the research through data and process triangulation. For instance, documents can be treated as underlining knowledge to support or dispute findings from interview when the case is analysed (Silverman, 2006; Yin, 2009).

The main sources of the data of this research are semi-structured interviews and documentation. Through semi-structured interview the members of the project team were able to share their subjective views about the project and their understandings of why and how requirement risks emerged in the project. The total number of five members of the project team who played significant role in the project was taken part in this study including project coordinator, project manager, project leader, system analyst, and project developer (Table 2).

Position Held in the Project	Years of Experience in this Company	Hours of Interview	Data Collection Period
Project Coordinator	5	1.05	October 2008
Project Manager	5	2.05	October 2008
System Analyst	5	2.05	October 2008
Project Leader	5	0.55	October 2008
Project Developer	2	2.05	October 2008

Table 2: Details of the interview participants

Relevant project documents were collected with a purpose to triangulate interview data as well as to validate spelling of the project name, participants, and technical terms used in the project given by the interviewees. The documents collected include company profile, user organisation information, project background, system diagrams, and the project meeting minutes.

Based on the information given by the interviewees, thematic analysis was then employed. According to David and Sutton (2004), thematic analysis is an approach to help researchers to emphasise key issues leading to ability to investigate and explain phenomenon under study in a more structured way. By using this analytical approach,

the researchers were able to narrow their focus to specific points of interests reviewed in the theoretical background, i.e., three types of requirements risk.

3.2 The Case Study Background

The RF system is a pseudonym for a three million baht information system that was developed for a non-profit state enterprise under the authority of Thailand Ministry of Agriculture and Cooperative. The objective of the RF project was to develop an electronic transaction processing system for the finance department, personnel department, and supply department in the organisation. This system was designed to enable the staff in central and provincial offices to access and process organisation's data electronically. The project team was expected to deliver this system within seven months after the project was launched. However, at the end of the delivery phase, the users refused to approve the system because the system could not fulfil their needs. The developer team was given extension to fix the problems. The project was three months late and nearly 100,000 bahts over the initial budget.

4.0 Analysis

4.1 Risks of Incomplete Requirement

Risks of incomplete requirements came from two sources: users and management structure in the organisation. The data suggests that users' resistance to a new system, lack of knowledge of the existing system, and fear of losing face in front of others were the reasons that incomplete requirements occurred. The project team found that it was difficult to reach agreements with users on the requirements for the new system. They believed that resistance to the new system prevented users from giving full details of system requirements. For instance, a project developer pointed out *"When we analysed requirements given by the users, we found that it was very difficult to reach a conclusion because they were not keen to learn new technology."*

The lack of knowledge of the existing system among the staff might be able to explain why the staff could not articulate their requirements based on the existing system. The project leader believed that *"[...] Users knew the system requirements but did not actually understand the requirements in details"*. Likewise, the project manager found that *"[...] Older users did not know the system that well. In fact, their knowledge of the current system is limited. They understood only the features that they frequently use in daily operations."* Users' fear of losing face in front of others was another reason that the project team believed contributing to the incomplete requirements. It

was evident that the key users who were expected to give system requirements to the project team were reluctant to give information about the existing system and expectations for the new system.

In addition, the management structure in the user organisation prevented users from giving full details about their requirements. As the project leader stated “[...] *the government agencies are all the same. Their users were reluctant to make any decisions or say anything without permission from their superiors.*” For example, the head of procurement unit in the supply department assigned some of her staff in the unit to collaborate with the project team because she did not have much knowledge of technical details of the system. But since the assigned staff was not given permission to make any decisions about the new system, the actual requirements remained uncertain. As the result, the project team followed the suggestions given by the most senior user staff in the procurement unit. The project leader stated that “*This person had been working in this unit for nearly 30 years. [...] He was thus considered (by us) as an expert in this area (to provide the requirements).*”

4.2 Risks of Misunderstanding Requirements

The problem of project team misunderstanding the system requirements only became apparent to the project team when the end users rejected the RF system. Before demonstrating the RF system to the users, the project team was confident with the system and believed that the only thing left to do was to run training sessions for the users in the central and provincial offices. However the users from the provincial offices rejected the new system and argued that the functionalities in the new system especially those ones in the supply system do not match the actual practices.

The interviewees admitted that part of this problem came from their confidence in the accuracy of requirements given by the users in the central office and failed to verify the requirements with the users in provincial offices. As project leader said “*We knew that the staff in the provincial office will use this system too and perhaps should be involved, but the users in the central office convinced us that provincial offices must follow the central office’s recommendations. They even argued that the project would be more problematic if we involved staff from the provincial offices in the requirement collection stage.*” System analyst added that “*Staff in the central office told me that they used to work in provincial offices and therefore the requirements that they gave us show the actual workflows of the system used in the provincial offices.*”

As the result, the project team had to redesign the system by taking into account the requirements given by the users in the provincial offices. The supply system for the procurement unit has to be redesigned and redeveloped from scratch. *“For this subsystem, I’d say out of five phases of development, we had to redo it from phase two.”* (Project developer)

4.3 Risks of Changing Requirements

Changing requirements took place in the extension phase. The main causes of changing requirements at this stage of the project were conflicts between user groups and changing project champion. There were conflicts between users from different units in the supply department regarding the requirements. For example project developer observed that users from one unit had tried to shift some of their responsibilities within the current system to users in the other units. *“[...] Users from each unit in the supply department tried to avoid having responsibility for the process, e.g. receipt issuing system, which is currently shared among units in the department. They even asked us if it is possible to transfer this responsibility to other units in the new system.”*

The project team also noticed that there were disagreements among male and female user groups over user requirements in the supply department. For example, the male user group was satisfied with the new system being implemented and thus no modification was required while the female user group asked the system developers to redesign the system. The conflicts among user groups led to further changes in the system requirements and created uncertainty to the project as a whole. The project team had to turn to the project champion (director of information centre) for help with solving the conflicts. The conflicts were eventually solved after several meetings held by the project champion.

Changing project champion from the director of information centre to the director of supply department in the extension phase contributed to further modification of user requirements. The incident of change of project champion took place towards the end of the system development. *“The director of information centre had an argument with the head of procurement unit. [...] As far as I heard, the director was blamed for taking the developer company’s side and not protecting the organisation’s benefits. In fact, we believed that the director had done everything to assure that his organisation*

will have all benefits from this project, but the director of supply department had already come to a conclusion about him.” (Project leader)

The new project champion was the director of supply department while the previous project champion (the director of information centre) was asked to continue collaborating with the project team to develop the rest of the project. The result of this change delayed project delivery because the director of supply department was uncertain about how the project should be approved. Therefore, she asked her subordinates and the director of information centre to assist in the system approval. Additional requirements were added to the original requirements after thorough checks by the users and the director of information centre in this phase. This created project complexity and ultimately delayed the project delivery. For instance, project leader noticed that *“After we demonstrated the final system, we thought that we were done. But, they (users from the supply department) spent a lot of time on testing and adding more requirements to the system.”* Moreover, the team realised that the director of supply department was in fact not ready for the system delivery and tried to delay this process. Project developer noticed that *“when the time came they said that they need more time to approve the project. They also admitted that they have not yet hired a security company to watch computer equipments previously installed in their organisation.”*

The problem of changing requirements occurred repeatedly in the extension phase. The situation reached to the point that top management of the developer company ordered the project team to finish the project as soon as possible, otherwise the company would not only face with financial difficulties but also risk its reputation in the sector. The project leader thus used a checklist of the remaining requirements mutually agreed between the project team and the users. By using the checklist, the project team were able to finalise user requirements and complete the project. Table 3 summarises requirement risks identified, the reasons that they occurred, and the risks that they induced in the project.

Phase of Development	Requirements Risk	Nature of Problem	Influential Reason	Consequence
Requirement collection and analysis	Incomplete Requirements	Users were reluctant to give comments about the system	Lack of knowledge of the existing system, users resistance to the new system, the fear of loosing	The project team spent more time on recollecting requirements

			face in front peers	
Requirements collection and analysis	Incomplete Requirements	Users did not want to make any decision regarding the system being implemented	Management structures in the user organisation	No definite decision about user requirements was made. The project team therefore followed the requirements given by a senior member of staff. This subsequently contributed to user rejection.
System Training	Misunderstanding Requirements	End-users rejected the system because the system was not developed according to the actual users' requirements	Project team over relied on requirements given by users from the central office and failed to verify the requirements with users in provincial offices	The project was requested to redesign some parts of the system
Project Extension (Delay)	Changing Requirements	Disagreement about requirements between users	Users were in competitive working environment	Project team had to ask project champion to solve the problems.
Project Extension (Delay)	Changing Requirements	Requirements changed repeatedly	Change of project champion from a person who strongly supported the project to another person who did not know much about the project	Requirements changed frequently as a result the project team had to use a checklist of the remaining requirements agreed by itself and the users.

Table 3: Summary of requirement risks in RF project

5.0 Discussion

It is evident from the case study that requirement risks can be found in different stages of an IS project other than in requirement collection and analysis. Requirement risks can be found in different forms and reasons for requirement risks are usually complex.

It is also observed in the case study that one form of requirement risk can lead to another form of requirement risk.

To avoid losing face in front of others, users may be inclined to provide only superficial requirements and comments about the system. It was found that because of users' face-saving behaviour many requirements were unstated, the project was delayed, and the initial cost of project was escalated. This study identifies a linkage between the incidents of face-saving and 1) the lack of required knowledge about the system and technology, and 2) the age of the users who provided requirements. In particular, this project demonstrated that users had face-saving characteristics because they lacked of required knowledge and old.

The management structures in an organisation can influence user commitment to provide requirements. The case study illustrates a situation where users were reluctant to give definite requirements or make decisions because of the hierarchy in the organisation. The hierarchy in the organisation means that all requirements given by the users would have to be approved by the management unless the management gave permissions to the users to make the decisions. The management structure might also have impacts on users' motivations to take active role in giving requirements. For instance, the project team wondered if the users were forced to collaborate with them to give requirements. He noticed that "*[...] they (users from the supply department) were quite passive. I think that their superior must have forced them to participate in these meetings. [...] they always said yes to all the questions asked.*" This inevitably created problem for requirement collection as the project team was unable to collect sufficient details of the requirements. Having incomplete requirements means that the project team had to spend more time and effort to looked information somewhere else and which subsequently led to risk of misunderstanding requirements.

The misunderstanding occurred was partly because of incomplete requirements which led the project team to rely on one single source of information. Because the information came from a senior member of staff the project team was over confident with the accuracy of the requirement and failed to verify them with other users. Another reason might have been that the project team developed the system based on their own interpretations of the requirements without confirming their understandings with the users. This is not uncommon in ISD project as Pan et al. (2004) find that developers may simply build IS projects based on their assumptions without verifying their understandings with the end-users. The worse outcome of misunderstanding

requirements was observed in this case study, that is, the system was developed on the basis of wrong assumptions and therefore rejected by the users.

The study found that conflict among user groups is the source of continuing changes in requirements. The conflict may come from competition among user groups in order to protect their own interests. A possible outcome of such competition is that the original objectives could be diverted (Teger, 1980) or escalation of project (Rubin et al., 1980; Newman and Sabherwal, 1996). For instance, because of competing interest users from different units in the supply department tried to defend their own requirements against others' requirements. As the result, the requirements were changed all the time and it was difficult to achieve consensus on the requirements for the final system. This situation may happen when users in the different departments have different project goals and deliverables which subsequently lead to uncertainty in the project development (Schmidt et al., 2001). It is also interesting to note that the differences in opinions of requirements may be due to gender difference. The female users in this study seemed to pay more attention to the details of the system features than the male users. The former type of users asked for changes of the system features while the latter type of users were happy to accept the system given by the project team.

Changing project champion induced to changes in user requirements in this study. This finding is in line with other studies which also suggest that continuity of project champion contributes to continuity of commitment on the same course of action whereas discontinuity of project champion could result a withdrawal of project commitment (Keil, 1995; Montealegre and Keil, 2000). Montealegre and Kiel (2000) find that change of project champion leads to reconsideration of resource allocation in order to prevent a project to continue on the same direction. Replacing project champion in the later stage of an IS project with someone who is less familiar with the project can also induce changes in requirement and create uncertainty in the project. For example, the new project champion tried to postpone project approval because she was unsure about the project requirements and was not ready for the project delivery. As project leader expressed "*they (director of supply department and her staff) and us do not speak the 'same' language. Hence, I thought that it might have been better if the project champion was an IT person e.g. director of information centre who can coordinate with users for us.*"

6.0 Conclusion

Before concluding this study, it is necessary to discuss some of its key limitation. First, in practice, there are many requirements risks in IS projects; however, not all requirements risks that can occur in IS projects were identified and thus investigated in this research. Second, the IS project chosen for this study was considered as having budget overrun and late delivery problem. Requirement risks and the reasons for the emergence of these risks in this project may therefore be different from successful projects. Hence it is necessary to carry out further research on requirement risks in successful IS projects to compare and contrast the results of this study.

Requirement risk has been identified and known as a significant factor that contributes to IS project failure. However, little research has focused on the emergence of requirement risks. This study therefore investigates the situation where requirement risks emerge. The outcome of this study not only confirms the results of the previous studies but also offers additional insights into the understandings of requirement risks in three different types and their occurrences. The findings suggest that reasons for requirement risks are complicated and involve with user commitment, management structures, and politics in the organisations. The findings enrich our previous understanding of problems of gathering user requirements (e.g. users' inability to articulate their requirements).

The results of this study contribute to the current understandings of emergence of requirement risks in IS projects. By understanding the roots of requirements risks, IS practitioners may be able to avoid the potential drawbacks of requirement risks and alleviate the degree of IS project complexity caused by the risks. For instance, project developers may have to verify their understandings of requirements with the end-users in the requirement collection phase even if the requirements are given by the reliable person, or the project developers must identify and solve possible conflicts regarding user requirements and project scope between user units in the user organisation before the requirements are implemented, etc. In term of future research, researchers can build on these findings to evaluate requirement risks in other IS projects that are in different sizes or have different outcomes.

Reference

Addison, T. (2003) *E-commerce Project Development Risks: Evidence from a Delphi Survey*. International Journal of Information Management, 23, 25 - 40.

- Ayoo, P.O. & Lubega, J.T. (2009) *A Requirements Analysis Framework for Human Activity Systems (HAS): The Case of Online Learning*. In: Kizza, J.M., Lynch, K., Nath, R., Aisbett, J. & Vir, P. (eds.), *Strengthening the Role of ICT in Development*, Vol. V, pp. 14 - 34. Oxford, UK: Fountain Publishers Limited.
- Barki, H., Rivard, S. & Talbot, J. (2001) *An Integrative Contingency Model of Software Project Risk Management*. *Journal of Management Information Systems*, 17 (4), 37 - 69.
- Bleistein, S.J., Cox, K., Verner, J.M. & Phalp, K.T. (2006) *B-SCP: A Requirements Analysis Framework for Validating Strategic Alignment of Organizational IT Based on Strategy, Context, and Process*. *Information and Software Technology*, 48 (9), 846 - 868.
- Boehm, B. (2000) *Project Termination Doesn't Equal Project Failure*. *Computer*, 33 (9), 94 - 96.
- Borland Software Corporation (2005) In A Borland White Paper Borland Software Corporation, Austin, TX, pp. 1 - 14.
- Carter, R.A., Anton, A.I., Laurie, W. & Dagnino, A. (2001) *Evolving Beyond Requirements Creep: A Risk-based Evolutionary Prototyping Model*. *Proceeding of the Fifth IEEE International Symposium on Requirements Engineering*, 94 - 101.
- Creswell, J.W. (2007) *Qualitative inquiry and research design: Choosing among five approaches*. Thousand Oaks, CA: Sage.
- Creswell, J.W. (2009) *Research design: Qualitative, quantitative, and mixed methods approaches*. Thousand Oaks, CA: Sage.
- Daft, R.L. & Macintosh, N.B. (1981) *A Tentative Exploration into the Amount and Equivocality of Information Processing in Organizational Work Units*. *Administrative Science Quarterly*, 26, 207 - 224.
- David, M. & Sutton, C., D (2004) *Social research*. London: Sage.
- DeMarco, T. & Lister, T. (2003) *Risk Management During Requirements*. *IEEE Software*, 20 (5), 99 - 101.
- Eisenhardt, K.M. (1989) *Agency Theory: An Assessment and Review*. *Academy of Management Review*, 14 (1), 57 - 74.
- Fowler, M. (2001) *The New Methodology*. *Wuhan University Journal of Natural Sciences*, 6 (1 - 2), 12 - 24.
- Glass, R.L. (2001) *Frequently Forgotten Fundamental Facts about Software Engineering*. *IEEE Software*, May/ June, 110 - 112.
- Gottesdiener, E. (2009) *How Agile Practices Reduce Requirements Risk*, Vol. July/ August EBG Consulting, Sudbury, MA.
- Hall, T., Beecham, S. & Rainer, A. (2002) *Requirements Problems in Twelve Companies: An Empirical Analysis*. *IEEE Proceedings Software*, 149 (5), 153 - 160.
- Han, W.-M. & Huang, S.-J. (2007) *An Empirical Analysis of Risk Components and Performance on Software Projects*. *The Journal of Systems and Software*, 80, 42 - 50.
- Hickey, A.M. & Davis, A.M. (2004) *A Unified Model of Requirements Elicitation*. *Journal of Management Information Systems*, 20 (4), 65 - 84.
- Hodkinson, P. & Hodkinson, H. (2001) In *Learning and Skills Development Agency Conference* Cambridge, UK.
- Howcroft, D. & Wilson, M. (2003) *Paradoxes of Participatory Practices: the Janus Role of the Systems Developer*. *Information and Organization*, 13, 1 - 24.

- Jones, C. (1996) *Strategies for Managing Requirements Creep*. Computer, 29 (6), 92 - 94.
- Kasser, J. (2002) *A Prototype Tool for Improving the Wording of Requirements*. The 12th Annual International Symposium of the INCOSE Las Vegas, NV. Las Vegas, NV.
- Keil, M. (1995) *Pulling the Plug: Software Project Management and the Problem of Project Escalation*. MIS Quarterly, 19 (4), 421 - 447.
- Keil, M., Cule, P.E., Lyytinen, K. & Schmidt, R.C. (1998) *A Framework for Identifying Software Project Risk*. Communications of the ACM, 41 (11), 76 - 83.
- Keil, M. & Mann, J. (1997) *Understanding the Nature and Extent of IS Project Escalation: Results from a Survey of IS Audit and Control Professionals*. Proceedings of the Thirtieth Annual Hawaii International Conference, 1-10.
- Keil, M., Mann, J. & Rai, A. (2000) *Why Software Projects Escalate: An Empirical Analysis and Test of Four Theoretical Models*. MIS Quarterly, 24 (4), 631 - 664.
- Kim, H., Iijima, J. & Ho, S. (2005) *A Framework for Analysis of System Failure in Information Systems Integration*. Industrial Engineering & Management Systems, 4 (2), 207 - 217.
- Kumar, R.L. (2002) *Managing Risks in IT Projects: an Options Perspective*. Information and Management, 40, 63 - 74.
- Lauesen, S. & Vinter, O. (2001) *Preventing Requirement Defects: An Experiment in Process Improvement*. Requirements Engineering, 6, 37 - 50.
- McAllister, C.A. (2006) *Requirements Determination of Information Systems: User and Developer Perceptions of Factors Contributing to Misunderstandings*. Capella University.
- McConnell, S. (1996) *Rapid Development: Taming Wild Software Schedules*. Redmond, WA: Microsoft Press.
- McEwen, S. (2004) *Requirements: An Introduction* [Online]. Metasys Technology, Inc. http://www.ibm.com/developerworks/rational/library/4166.html?S_TACT=105AGX78&S_CMP=HP [Accessed Nov 19, 2009]
- Montealegre, R. & Keil, M. (2000). *De-escalating Information Technology Projects: Lessons from the Denver International Airport*. MIS Quarterly, 24 (3), 417 - 447.
- Newman, M. & Sabherwal, R. (1996). *Determinants of Commitment to Information Systems Development: A Longitudinal Investigation*. MIS Quarterly, 20 (1), 23 - 54.
- Nidumolu, S.R. (1996). *Standardization, Requirements Uncertainty and Software Project Performance*. Information and Management, 31, 135 - 150.
- Oberg, R., Probasco, L. & Ericsson, M. (2000) Rational Software Corporation, pp. 1 - 24.
- Pan, G.S.C., Shan, L.P. & Flynn, D. (2004) *De-escalation of Commitment to Information Systems Projects: A Process Perspective*. Journal of Strategic Information Systems, 13, 247 - 270.
- Pare, G., Sicotte, C., Jaana, M. & Girouard, D. (2008) *Prioritizing Clinical Information System Project Risk Factors: A Delphi Study*. Proceeding of the 41st Hawaii International Conference on System Sciences, 1 - 10.
- Robinson, W. N., Pawlowski S. D., and Volkov, V. (2003) *Requirements Interaction Management*, ACM Computing Surveys, 35, pp. 132 - 190.
- Rubin, J.Z., Brockner, J., Small-Weil, S. & Nathanson, S. (1980) *Factors Affecting Entry into Psychological Traps*. Journal of Conflict Resolution, 24, 405 - 426.

- Schware, R. & Bhatnagar, S. (2001) Thailand's Troubled Tax Computerization Project. The World Bank's Information Solutions Group. (e-Government Practice Report).
- Sabherwal, R., Jeyaraj, A. & Chowa, C. (2006) *Information System Success: Individual and Organizational Determinants*. Management Science, 52 (12), 1849 - 1864.
- Schmidt, R., Lyytinen, K., Keil, M. & Cule, P. (2001) *Identifying Software Project Risks: An International Delphi Study*. Journal of Management Information Systems, 17 (4), 5 - 36.
- Shull, F., Rus, I. & Basili, V. (2000) *How Perspective-based Reading Can Improve Requirements Inspections*. Computer, 33 (7), 73 - 79.
- Silverman, D. (2006) *Interpreting Qualitative Data*. Thousand Oaks, California: Sage.
- Stake, R.E. (1995) *The art of case study research*. Thousand Oaks, CA: Sage.
- Staw, B.M. & Ross, J. (1987) *Behaviour in Escalation Situations: Antecedents, Prototypes, and Solutions*. Research in Organizational Behavior, 9 (40), 39-78.
- Sumner, M. (2000) *Risk Factors in Enterprise Wide Information Management Systems Projects*. SIGCPR, Evanston, IL. Evanston, IL.
- Teger, A. (1980) *Too much invested to quit*. New York: Pergamon.
- The Standish Group (1995) *The CHAOS Report*.
- Thong, J.Y.L., Yap, C. & Raman, K.S. (1996) *Top Management Support, External Expertise and Information Systems Implementation in Small Businesses*. Information Systems Research, 7 (2), 248 - 267.
- Tiwana, A. & Keil, M. (2004) *The One-minute Risk Assessment Tool*. Communications of the ACM, 47 (11), 73 - 77.
- Verner, J.M., Cox, K., Bleistein, S. & Cerpa, N. (2005) *Requirements Engineering and Software Project Success: An Industrial Survey in Australia and the US*. Australian Journal of Information Systems, 13, 225 - 238.
- Wieggers, K.E. (2000) *Karl Wieggers Describes 10 Requirements Traps to Avoid*. Software Testing & Quality Engineering, January/ February 2000, 1 - 8.
- Wieggers, K.E. (2003) *Software Requirements*. Redmond, WA: Microsoft Press.
- Williams, D. & Kennedy, M. (1999) *A Framework for Improving the Requirements Engineering Process Effectiveness*. International Council on Systems Engineering Conference, June 5 - 11, Brighton, UK. Brighton, UK.
- Yin, R.K. (2009) *Case Study Research: Design and Methods*. Thousand Oaks, CA: Sage.