

Association for Information Systems AIS Electronic Library (AISeL)

CONF-IRM 2011 Proceedings

International Conference on Information Resources
Management (CONF-IRM)

6-2011

Managing User Requirement Risks – An Exploratory Study of IS Projects from the Views of Outsourcing Teams

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

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

Follow this and additional works at: <http://aisel.aisnet.org/confirm2011>

Recommended Citation

Parinyavuttichai, Nipon and Lin, Angela, "Managing User Requirement Risks – An Exploratory Study of IS Projects from the Views of Outsourcing Teams" (2011). *CONF-IRM 2011 Proceedings*. 16.
<http://aisel.aisnet.org/confirm2011/16>

This material is brought to you by the International Conference on Information Resources Management (CONF-IRM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in CONF-IRM 2011 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Managing User Requirement Risks – An Exploratory Study of IS Projects from the Views of Outsourcing Teams

Nipon Parinyavuttichai, Angela Lin
The University of Sheffield
N.Parinyavuttichai@Sheffield.ac.uk, A.Lin@Sheffield.ac.uk

Abstract

Requirement risk is often identified as a reason contributing to Information Systems Development (ISD) project failure but has not been adequately explored in the IS literature. Requirement risk refers to the uncertainties caused by differences between the requirements perceived by system developers and user groups' expectation of the system. Requirement risk can lower the project performance therefore effective management of requirement risk is critical to achieve desired project performance. This paper aims to address the gap in the current literature of requirement risk by emphasising its dynamic nature and examining the strategies to manage such risks. This study identifies three types of requirement risk: changing requirement, misunderstanding requirement, and incomplete requirement and argues that each type if not being attended to by the project team will lead to further requirement change or other project risks. The study also identifies the strategies that a project team can employ to manage requirement risk.

Keywords

Requirement risk, socio-organisational perspectives, requirement determination, IS failure, emergence, ISD, risk management

1. Introduction

Information systems (IS) development is a highly complex process. Many ISD projects failed to deliver on time, within budget, and with required functionalities, and some even had to be cancelled (Boehm 2000; Barki, Rivard et al. 2001). Among various reasons for ISD project failures (Doherty, King et al. 2002; Jackson and Klobas 2008) requirement risk has been frequently mentioned but inadequately discussed (McEwen 2004; Verner, Cox et al. 2005). The prior research on requirement risk tends to focus its attention on identifying different types of requirement risk (Shull, Rus et al. 2000; Han and Huang 2007) and factors that cause them (Wiegers 2000; McAllister 2006). The research somewhat subscribes to a snap shot view that regards requirement risk as a consequence of actions taken or not taken in the early stage of an ISD project. However requirement risk can occur at any stage of an ISD project and if it is not managed appropriately and in time it can lead to other project risks

(Parinyavuttichai and Lin 2010). Therefore it is important to acknowledge that requirement risk is dynamic in nature and strategies need to be in place to eliminate and/or control the risks throughout the project. This study aims to fill this gap in the current understanding of requirement risk by demonstrating the dynamic nature and examining the management strategies employed to manage the risks.

2. Theoretical Background

2.1. Requirement Risk

Requirement risk of an IS project refers to the uncertainties caused by differences between the requirements perceived by systems developers and user groups' expectations of the system (Daft and Macintosh 1981). The types of requirement risk that can be found in most IS projects are changing requirement risk, misunderstanding requirement risk, and incomplete requirement risk (Parinyavuttichai and Lin 2010). Changing requirement risk is usually the outcome of significant additions or modifications to the system requirements throughout an IS project (Carter, Anton et al. 2001). The common causes of changing user requirements include changing needs due to technological or business changes (Land 1982), conflicts between user departments (Teger 1980), lack of the understanding of the system in development among users (Kumar 2002), and the choice of ISD methodology (Carter, Anton et al. 2001). Changing user requirements can irritate systems developers because of the uncertainties caused by it (Carter, Anton et al. 2001) and can lead to escalation of development costs (Tiwana and Keil 2004; Paré, Sicotte et al. 2008). Although it is possible to use prototyping methods to reduce changing user requirements caused by the lack of the understanding of the system among users, it is not always possible to avoid the changes that stem from outside the organisation (Fowler 2001).

Misunderstanding requirement risk refers to the situations where system developers and users have different expectations of an information system. These situations arise because neither developers nor users have a clear understanding of the system requirements (Dey, Kinch et al. 2007; Gottesdiener 2009), there are miscommunications between users and systems developers (Coughlan, Lycett et al. 2003), or different worldviews are brought to the creation of an IS by users and systems developers (Wiegiers 2000; Kudikyala and Vaughn 2005). Misunderstanding requirements risk can lead to subsequent project development problems which consists of changing requirements, changing system design, user dissatisfaction, higher project cost, and even delay in project delivery (Wiegiers 2003; McAllister 2006; Gottesdiener 2009).

Incomplete requirement risk occurs when some user requirements are ignored or overlooked by the project team or users themselves (Lauesen and Vinter 2001). For instance, users may change their expectations of an information system over time (Hecht and Hecht 2000); a project team may inappropriately use evaluation tools to capture user requirements (Hecht and Hecht 2000); or a project team may have developed the system based on wrong assumptions without verifying their understanding against end-user expectations (Howcroft

and Wilson 2003). Incomplete requirement risk leads to ISD complexities (Na, Simpson et al. 2007). For example, project teams may have to spend more time and project resources to collect more requirements in the later phases (Lauesen and Vinter 2001); or some errors or subsequent project problems are hidden in the requirements that are not yet collected (Hecht and Hecht 2000). For these reasons, a project team could spend up to 80% of the project team's effort to correct the problems arising from the incomplete requirement risks (Williams and Kennedy 1999) therefore the early detection and management of incomplete requirements risks in the project is important to project success (Hecht and Hecht 2000).

2.2. Management of Requirement Risk

Requirement risk can lower the project performance therefore effectively managing requirement risk is critical to achieve desired project performance (Han and Huang 2007). Broadly speaking IS risk management can be categorised into two types: risk reduction and risk hedging (Kumar 2002). Risk reduction management aims to reduce causes of project uncertainties and it takes a snap shot view that risk management is an instantaneous activity. For example, a technique to reduce the project uncertainty caused by incomplete user requirements is to collect requirements from multiple stakeholders. Such views however do not take into account the fact that project risks are usually dynamic in nature and may be evolved and/or changed over time (Parinyavuttichai and Lin 2010). Developing risk hedging strategies to minimise the negative impacts of risk thus seems appropriate where risks cannot always be predictable and completely eliminated (Kumar 2002; Tiwana and Keil 2004). In this sense, risk management is a continuous activity and requires constant efforts from the stakeholders throughout a project (Stoneburner, Goguen et al. 2002). Table 1 summarises that the strategies that have been identified by the previous literature to manage requirement risks.

3.1. Research Strategy

As the aim of this study is to understand the emergence of requirement risks and the strategies developed to manage the risks over time a qualitative case study approach was employed. The qualitative case study approach is appropriate for the context of this study because the phenomenon of IS risks and risk management is unique in each project and the approach allows the participants to talk about their personal experience and provide their own views regarding the topic (Creswell, 2009; Yin, 2009). This study presents the findings of three case studies conducted in software companies in Thailand. The study mainly focuses on project risk management on the software companies' part. It is because while many established studies have made substantial contributions to risk management from user organisations' perspective (Jiang et al., 2000; Balaji et al., 2006) little is known how software companies manage project risks when interacting with user organisations. Table 2 summarises key information of three case studies under investigation.

Strategies	Definition	Reference
Effective communication and collaboration	Use of effective communication with the users as a way to strengthen user collaboration and increase user involvement	Bostrom (1989); Gallivan and Keil (2003); Hwang and Thorn (1999); Newman and Sabherwal (1996);
Technological management	Use of technology, tools, or standard software infrastructure to evaluate, assess, and/or resolve IS risks, e.g., system prototypes.	Baskerville and Stage (1996); Carter et al. (2001); Grønhjæk (1990); Hecht and Hecht (2000).
Use of knowledge and experience	Knowledge and experience of project management and system development among the project team members	Boehm (2000); Han and Huang (2007); Kudikyala and Vaughn (2005); Wiegers (2000);
Finding an alternative source of information	Seeking another person or group who can provide the project team the user requirements instead of the users	Howcroft and Wilson (2003); Müller and Turner (2005); Pan et al. (2004)
Political support	An attempt to engage the top management of the user organisations to resolve requirement situations	Han and Huang (2007);

Table 1: Key Strategies to Alleviate Requirement Risks

Characteristics	Project Code		
	NLT	RPAF	e-Paperless
Type of user organisation	A branch of Thai national library	Non-profit organisation under supervision of Ministry of Agriculture	Thai Customs
Objective(s) of the project	Implementation of Library Information Systems	Development of an electronic transaction processing system for finance, personnel, and supply department	Enhance trade facilitation system in order to ensure that the new system is compliant with the WCO standard
Project budget (millions baht)	3.5	3.0	228
Project duration (months)	7	7	19
Project outcome	On schedule; slightly cost escalation	Four months delayed; Cost overrun	Successfully developed

Table 2: Key Information of the Projects in This Study

3. Research Methodology

The principal sources of data are interviews and documentation. Semi-structured interviews were conducted to ensure the uniformity of the topic of interest or key themes across

interviews. In addition, it allows flexibility in adapting the questions to reflect the uniqueness of each case (del Barrio 1999). Table 3 summarises the details of the interviewees in each case study. The documentations including the profiles of the software companies, information about user organisations, project backgrounds, system diagrams, and project meeting minutes were collected for supporting our understanding of the project and data triangulation. The data was analysed using thematic analysis which aims to identify key issues of interest (David and Sutton 2004).

Project	Position Held in the Project	Years of Experience in the Company	Interview Duration (Hours)	Data Collection Period
NLT	Project manager	10	1.00	October 2008
	Project leader	10	2.00	October 2008
	System analyst	5	1.00	January 2009
	Project developer	7	1.00	October 2008
RPAF	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
e-Paperless	Project manager	20	2.20	October 2008
	1 st project leader	10+	2.20 (1 st interview) 2.40 (2 nd interview)	October 2008 and January 2009
	2 nd project leader	10+	2.20	October 2008
	3 rd project leader	10+	2.20	October 2008
	System analyst	10+	2.40	January 2009
	Project developer	2	2.40	January 2009

Table 3: Details of the Participant’s Interview Data

4. Findings

4.1. Adoption of System Prototypes

Prototyping method was used across the three projects to reduce the uncertainties stemming from situations where users had problems articulating their requirements (changing requirement) and where neither users nor the system developers had enough knowledge and clear understanding of the new system (misunderstanding requirement). Systems prototype helped users learn more about the new systems so that they could have a clearer understanding of the new system and subsequently give better description of their requirements. In the case of NLT, the project leader recalled that

Before (the system prototypes were used), users were unable to imagine what they wanted from us...Once they saw the prototypes [...], they understood what requirements we would like to have from them.

In addition, system prototypes helped the developers avoid misunderstanding user requirements as well as helped them find the project direction. As the project manager in RPAF stated:

System analysts designed layouts of the designing system based on user suggestions. We did not use prototypes to identify every feature of the final system. But users could understand how the system would be developed because the prototypes had covered 60% of the final system features.

4.2. Use of Knowledge and Skills

In order to manage project risks effectively having an experienced project manager on board was identified as a key to managing risks. This is because experienced project managers can draw on their previous experiences to develop strategies to manage risks. The risk management knowledge and skills that project managers possess are particularly important where changing user requirement is inevitable due to changes within and outside the organisation. The project manager of NLT commented that “*users tend to have more requests and ask for changes in system features*”. The e-Paperless project manager believed that the previous experience with other projects had taught her the importance of managing changing requirement risk promptly as it can easily lead to subsequent project complexities. This explains why she used various tactics to prevent and mitigate risk including filtering the requirements for change and had set aside contingency funds for unforeseen problems. As the project manager stated:

[...] I would say this was due to my knowledge and experience of ISD. When I perceived that the changes might create significant project impact, i.e., project delay, I would not allow the changes to occur.

In contrast, inexperienced project managers could lead a project into situations where requirement risk can escalate and become unmanageable. This was observed in the RPAF project as the project manager admitted later:

I was unable to control the situations of requirements change and thus allowed the users to make changes of their system. This is entirely because I did not have enough experience and skills to prevent the system from changing.

4.3. Effective Communication and Negotiation

Effective communication and negotiation was a mechanism used to hedge the risks across the three projects. A cause of changing requirement in both RPAF and e-Paperless cases was frequent changes in personnel in the user organisations. For instance, in the e-Paperless project it was observed that “*the organisation frequently reshuffle their users. And there was a tendency of changing user requirements when the new users came on board.*” (Project manager)

In the case of RPAF the project developers also faced incomplete requirement risks because of the conflicts between user departments about who should have responsibility of certain operational tasks. The conflicts not only prevented a consensus over the user requirements to be achieved but also prevented the developers to collect requirements. The incomplete

requirement risk in this case contributed to changing requirements since nothing was agreed so everything was subject to further changes.

“[...] Users from each unit in the supply department tried to avoid having any responsibility for giving us their requirements especially of the receipt issuing system which is mutually used by every unit. Some users even asked us to transfer the future responsibilities for the system to the other units in the department.”

Project mediators played an important role in these cases by helping the project manager to bring the projects back on track. The project mediators were from the user organisations and had knowledge of both the new system as well as the organisations. The mediators were asked to facilitate the meetings between the system development team and the representatives of user groups and to help them reach agreements on the user requirements. In the case of e-Paperless the mediator even suggested how the system can be developed.

Frequent communication with users can reduce requirement risks. For instance, the project team in RPAF tried to communicate with some users constantly to avoid the situations where user requirements were overlooked or neglected (incomplete requirements). This was partly because the users found it difficult to articulate their requirements.

“Their knowledge of the current system was quite limited. They could only think of the features that they frequently used in daily operations.” (Project leader in RPAF)

“During system development phase, we often communicated with the users to make sure that they would not have any further modification to the system design.” (Project leader in NLT)

4.4. Reliance on Alternative Sources of Information

Using alternative sources of information can help the situation where collecting user requirements is difficult. In the case of RPAF, users were not always willing to take responsibility of giving user requirements unless their superior gave them instructions to do so. Besides this the users were often unconformable giving suggestions in front others. As a result, the project team was unable to collect user requirements and hence faced incomplete user requirement risk. The strategy employed to solve the problem was to use an alternative information source, namely a senior member of staff in the user group. It was believed that because of the time that this senior member served in the organisation he would have a good idea of the requirements. By consulting with the alternative source, the project team could proceed to the next stage of ISD by developing the system based on the requirements given by the source. Table 4 summarises risk management approaches and the outcomes of risk management.

5. Discussion

Requirement risk can occur at any stage of a project and the reasons for its occurrence are not always predictable. In this study we have seen that relocation of users caused changing requirement due to the inconsistencies in user expectations; users' inability to articulate their

requirements and organisational hierarchy which prevented users to express their views led to incomplete requirements and subsequently caused changing requirements; and conflicts between user departments meant that consensus over the user requirements was difficult to achieve. The study also shows that requirement risks would manifest unless they were well managed.

Risk Management	Risk Type	Risk Incidents	Case (Phase)	Outcome
Adoption of system prototypes	Changing requirements	Lack of ability to articulate user requirements	NLT (RC)	Users developed better understanding of the new system
		Users are uncertain about the expectations of the system	RPAF (RC)	
	Misunderstanding requirements	Lack of clear understanding of the user system	NLT, RPAF, e-Paperless (RC, SD)	
Knowledge and skills in managing the projects	Changing requirements	Users changed requirements throughout the project	NLT (SD, SA); RPAF (SA); e-Paperless (RC, SD)	Some project teams could avoid the changing requirement situations
Effective communication and Negotiation	Different types of risks	Change of the person who was responsible for requirement provision	e-Paperless (RC, SD)	Some requests for the requirement change by the new users were avoided
		Conflict among users from different departments	RPAF (SA)	
Reliance on alternative sources of information	Incomplete requirement	Users were reluctant to provide their requirements	RPAF (RC)	The project team was able to go through the requirement collection phase

RC: Requirement Collection Phase; SD: System Development Phase; SA: System Adoption Phase

Table 4: Risk Management of Requirement Risks

The project managers in this study employed different strategies to manage requirement risks according to the types of risk and their experience of project management. The findings of the study suggest that risk reduction or prevention strategies are not always appropriate as requirement risk cannot be always foreseen. Conventional ISD textbooks would suggest that collecting requirements from multiple stakeholders will help build a fuller and more comprehensive picture. This view is based on the assumptions that (1) project teams are able to access to all stakeholders' views (2) all stakeholders are able to articulate their requirements (3) requirements will remain the same throughout. In practice, it is difficult to implement such a strategy. The study shows that the project teams faced difficulties in accessing users to collect the requirements and even the teams that had access found it difficult to obtain a complete set of requirements from users. Prototyping method was adopted in all three cases to help user groups understand the new systems better so that they

were able to articulate their requirements; and through the systems prototypes the developers were able to clarify the requirements with the user groups (Grønhjæk 1990; Haughey 2010). In all three projects, system prototypes were used as a risk reduction strategy that was abandoned after the requirements were collected. This risk reduction strategy could lead to other requirement risks. For examples, without constant verification of user expectations through system prototypes, project teams may simply underestimate the impacts of requirements change, leading to project uncertainties (Hecht and Hecht 2000).

The other category of risk management strategies is risk hedging. The risk hedging strategies view that risks cannot be eliminated totally and aim to minimize the undesirable outcomes. Developing risk hedging strategies to minimize the negative impacts of requirement risk would require constant efforts to monitor and manage the risk and expertise to do so (Han and Huang 2007). In the e-Paperless case it was apparent that the presence of an experienced project manager is the key to developing risk hedging strategy because the experience enabled the manager to foresee the problems on the horizon and therefore had strategies in place to deal with the situations. On the contrary the inexperienced project manager could (e.g. in the RPAF project) become a risk factor that escalated the situations of requirements change.

Inadequate systems requirements specifications is argued to be the outcome of ineffective communication between systems developers and users (Bostrom 1989). Hence many established studies advocate the importance of user-developer communication in project success (Newman and Sabherwal 1996; Hwang and Thorn 1999; Gallivan and Keil 2003). Effective communication is regarded as an essential risk management mechanism that needs to be in place to reduce project uncertainties. The effective communication can be achieved through direct instructions between project team and user groups, and the study shows that using a mediator as an interface between the project team and user groups can also be an effective strategy when the mediator has knowledge of the new system and user organisation. The role of this mediator is to facilitate the dialogues between the specialists (e.g. developers) and non-specialists (e.g. users) and help both parties to understand each other better. In other words the mediator plays the role to bridge the knowledge gap between developers and users. However, it is shown also that use of a mediator to help user-developer communication would only work if a mediator is experienced, knowledgeable, and equipped with negotiation skills. Constant user-developer communication can help a project team to detect the sign of changing requirements and therefore manage it in time.

In the situation where collecting user requirements was difficult, looking for the information source beyond the project could be a solution to identify the missing information. Such a strategy can address the incomplete requirement risk but it could also lead to other requirement risk if the information collected from other sources was not checked. The RPAF project team over relied on and trusted the alternative source. Therefore they failed to check the requirements collected from the source with the end-users. Consequently the system was built on the misunderstanding of the requirement and was eventually rejected by the users.

6. Conclusion and Implications

Requirement risk can lower the project performance and it is not possible to be eliminated completely from any project. In order to effectively minimize the negative impacts of requirement risk on an IS project, understanding the source of requirement risk becomes an essential task. This study has identified three types of requirement risk: changing requirement, misunderstanding requirement, and incomplete requirement. Each type if not attended to by project team will lead to further requirement change or other project risks. The study also identified the strategies that a project team can employ to manage requirement risk including using system prototype to collect user requirements, having an experienced project manager as a key project resource, effective use of communication and negotiation channels, and using alternative information sources.

IS practitioners may benefit from adopting the risk management mechanisms identified in this study to alleviate or even prevent potential requirement risks. The future research may want to build on the findings of the study to examine requirement risks and risk management strategies in other IS projects with different development strategy (partial outsourcing, in-house development), in different sizes, or with project outcomes.

Due to the constraints on its length, this paper can only discuss the common requirement risks observed across the three cases. Therefore some requirement risks identified in the established studies were not discussed in this paper.

References

- Barki, H., S. Rivard, et al. (2001). "An Integrative Contingency Model of Software Project Risk Management." *Journal of Management Information Systems* **17**(4): 37 - 69.
- Baskerville, R. L. and J. Stage (1996). "Controlling Prototype Development through Risk Analysis." *MIS Quarterly* **20**(4): 481 - 504.
- Boehm, B. W. (2000). "Project Termination Doesn't Equal Project Failure." *IEEE Computer* **33**(9 (September)): 94 - 96.
- Bostrom, R. (1989). "Successful application of communication technique to improve systems development process." *Information and Management* **16** 279-295.
- Carter, R. A., A. I. Anton, et al. (2001). Evolving Beyond Requirements Creep: A Risk-based Evolutionary Prototyping Model. *The Fifth IEEE International Symposium on Requirements Engineering*: 94 - 101.
- Coughlan, J., M. Lycett, et al. (2003). "Communication Issues in Requirements Elicitation: A Content Analysis of Stakeholder Experiences." *Information & Software Technology* **45**(8): 525 - 536.
- Daft, R. L. and N. B. Macintosh (1981). "A Tentative Exploration into the Amount and Equivocality of Information Processing in Organizational Work Units." *Administrative Science Quarterly* **26**: 207 - 224.
- David, M. and C. Sutton, D (2004). *Social research*. London, Sage.
- del Barrio (1999). "The Use of Semistructured Interviews and Qualitative Methods for the Study of Peer Bullying ". from http://old.gold.ac.uk/tmr/reports/aim2_madrid1.html.
- Dey, P. K., J. Kinch, et al. (2007). "Managing Risk in Software Development Projects: A Case Study." *Industrial Management & Data Systems* **107**(2): 284 - 303.

- Doherty, N., M. King, et al. (2002). "The impact of inadequacies in the treatment of organizational issues on information systems development projects." Information & Management **41**(49 - 62).
- Fowler, M. (2001). "The New Methodology." Wuhan University Journal of Natural Sciences **6**(1 - 2): 12 - 24.
- Gallivan, M. and M. Keil (2003). "The user-developer communication process: a critical study." Information Systems Journal **13**: 37-68.
- Gottesdiener, E. (2009) How Agile Practices Reduce Requirements Risk. Better Software **July/ August**,
- Grønhjæk, K. (1990). "Supporting Active User Involvement in Prototyping." Scandinavian Journal of Information Systems **2**(1): 3 - 24.
- Han, W.-M. and S.-J. Huang (2007). "An Empirical Analysis of Risk Components and Performance on Software Projects." Journal of Systems and Software **80**(2007): 42 - 50.
- Haughey, D. (2010, October 11, 2010). "Reaping the Benefits of Good User Requirements with the Unified Modeling Language (UML)." from <http://www.projectsmart.co.uk/pdf/reaping-the-benefits-of-good-user-requirements.pdf>.
- Hecht, H. and M. Hecht (2000). How Reliable are the Requirements for Reliable Software? . Software Technology Conference. UT, USA.
- Howcroft, D. and M. Wilson (2003). "Paradoxes of Participatory Practices: the Janus Role of the Systems Developer." Information and Organization **13**: 1 - 24.
- Hwang, M. I. and R. G. Thorn (1999). "The effect of user engagement on system success: a meta-analytical integration of research findings." Information and Management **35**(4): 229 - 236.
- Jackson, P. and J. Klobas (2008). "Building knowledge in projects: A practical application of social constructivism to information systems development." International journal of project management **26**(4): 329 - 337.
- Kudikyala, U. K. and R. B. Vaughn (2005). "Software requirement understanding using Pathfinder networks: Discovering and evaluating mental models." Journal of Systems and Software **74**: 101-108.
- Kumar, R., L (2002). "Managing Risks in IT Projects: An Options Perspective." Information & Management **40**(1): 63 - 74.
- Land, F. (1982). "Adapting to changing user requirements." Information & Management **5**: 59-75.
- Lauesen, S. and O. Vinter (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. School of Business. Minneapolis, MN, Capella University. **Doctor of Philosophy**: 291.
- McEwen, S. (2004). "Requirements: An Introduction." Retrieved Nov 19, 2009, 2009, from http://www.ibm.com/developerworks/rational/library/4166.html?S_TACT=105AGX78&S_CMP=HP.
- Müller, R. and J. R. Turner (2005). "The impact of principal-agent relationship and contract type on communication between project owner and manager " International Journal of Project Management **23**(5): 398 - 403.
- Na, K.-S., J. T. Simpson , et al. (2007). "Software Development Risk and Project Performance Measurement: Evidence in Korea." Journal of Systems and Software **80**(4): 596 - 605.
- Newman, M. and R. Sabherwal (1996). "Determinants of commitment to information systems development: a longitudinal investigation." MIS Quarterly **20**(1): 23 - 54.

- Pan, G. S. C., L. P. Shan, et al. (2004). "De-escalation of Commitment to Information Systems Projects: A Process Perspective." Journal of Strategic Information Systems **13**: 247 - 270.
- Paré, G., C. Sicotte, et al. (2008). Prioritizing Clinical Information System Project Risk Factors: A Delphi Study. The 41st Hawaii International Conference on System Success. Hawaii, USA.
- Parinyavuttichai, N. and A. Lin (2010). Understanding the Emergence of Requirement Risks in Information Systems Projects. The 15th UKAIS Oxford, UK.
- Shull, F., I. Rus, et al. (2000). "How Perspective-based Reading Can Improve Requirements Inspections " Computer **33**(7): 73 - 79.
- Stoneburner, G., A. Goguen, et al. (2002). Risk Management Guide for Information Technology Systems. NIST. Falls Church, VA, NIST.
- Teger, A. (1980). Too Mush Invested to Quit. New York, Pergamon.
- Tiwana, A. and M. Keil (2004). "The One-Minute Risk Assessment Tool." Communications of the ACM **47**(11): 73-77.
- Verner, J. M., K. Cox, et al. (2005). "Requirements Engineering and Software Project Success: An Industrial Survey in Australia and the US." Australian Journal of Information Systems **13**: 225 - 238.
- Wiegiers, K. E. (2000). "Karl Wiegiers Describes 10 Requirements Traps to Avoid." Software Testing & Quality Engineering **January/ February**: 1 - 8.
- Wiegiers, K. E. (2003). Software Requirements. Redmond, WA, Microsoft Press.
- Williams, D. and M. Kennedy (1999). A Framework for Improving the Requirements Engineering Process Effectiveness International Council on Systems Engineering Conference. Brighton, UK.