Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 80 (2007) 596–605
The Journal of Systems and Software
www.elsevier.com/locate/jss

Software development risk and project performance measurement: Evidence in Korea

Kwan-Sik Na [a], James T. Simpson [b,*], Xiaotong Li [b], Tushar Singh [b], Ki-Yoon Kim [c]

[a] School of Business, Seowon University, 231 Mochung-dong, Hungduk-gu, Cheongju-shi, Chungbuk 361–742, Republic of Korea
[b] The College of Administrative Science, The University of Alabama in Huntsville, Huntsville, AL 35899, USA
[c] School of Business, Kwangwoon University, 447-1 Wolgye-dong, Nowon-gu, Seoul 139–701, Republic of Korea

Received 12 December 2005; received in revised form 19 June 2006; accepted 24 June 2006
Available online 8 August 2006

Abstract

As more US companies outsource their software projects overseas, they find that it is more challenging to control software development risk in countries with dissimilar IT capabilities. Using data collected from software projects developed in Korea, we investigate the impacts of specific risk management strategies and residual performance risk on objective performance measures such as cost and schedule overrun. Our results indicate that, unlike subjective performance measures, our objective cost and schedule overrun performance measures are positively associated with residual performance risk in Korea. We also investigate the impact of two alternative conceptualization of software development risk on both objective performance and subjective performance. Finally, we discuss relevant policy implications for software development and outsourcing.

Keywords: Capability Maturity Model Integration; Project performance; Risk management; Software development risks; Software outsourcing; Standardization

1. Introduction

Software project management has received considerable attention from both academics and practitioners. This attention stems in part from the frequent reports of high-profile cases of mismanaged software development projects (Charette, 2005). In fact, many of the significant cost overruns occur in companies that specialize in commercial software development. These unanticipated overruns typically absorb valuable resources and occasionally necessitate abandoning a project altogether. In a highly publicized story of software failure in 2005, the FBI decided to scrap its $170 million Virtual Case File (VCF) software project. This move resulted in overruns of over $100 million which prompted several congressional hearings (Goldstein, 2005).

While there are many factors that may result in software failure, inadequate risk control is undoubtedly one of the leading factors (Barros et al., 2004; Tiwana and Keil, 2004). The extant IT literature has produced a number of conceptual frameworks to explain different types of software development risk, risk management strategies, and measures of software project performance (e.g., Fairley, 1994; Chittister and Haimes, 1996; Nidumolu, 1996a; Wallace and Keil, 2004). Most of these studies address software risk control in developed countries with sophisticated software development infrastructures. Because of the substantial increase in software development outsourcing to firms overseas, we believe that it is also important to have an in-depth understanding of software risk management practices in countries with less advanced software development infrastructures. For instance, a recent report revealed that three-quarters of US companies outsourced some of

* Corresponding author. Tel.: +1 256 824 6408; fax: +1 256 824 6328.
E-mail addresses: ksna@seowon.ac.kr (K.-S. Na), simpsonj@uah.edu (J.T. Simpson), lixi@uah.edu (X. Li), min1203@daisy.kwangwoon.ac.kr (K.-Y. Kim).

0164-1212/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2006.06.018

their IT activities in 2004, and that the percentage is expected to continue to climb in the near future (Minevich and Richter, 2005). Despite the strong momentum of global outsourcing, many of the world's largest companies are reporting increased dissatisfaction with their IT outsourcing outcome. Firms often find that as the complexity of the outsourcing transactions increase, it becomes more difficult for them to understand their offshore partners' business processes and management practice (Deloitte Consulting, 2005; Weakland, 2005).

Several important studies (i.e., Nidumolu, 1996a,b) have reported that software development risk is an important predictor of software performance. Yet, most IT studies have evaluated IT projects in mature IT organizations located in the United States. Only Na et al. (2004) have conducted a comparative study to determine how risk management strategies and residual risk impact software product and process performance in countries with dissimilar IT capabilities. Na et al. (2004) found that models developed with data collected in IT developed countries (i.e., Nidumolu, 1996b) do not apply to organizations in Korea, an IT developing country where the IT capability is known to be lower than in the US. Specifically, their study suggests that, unlike studies conducted in the US, residual risk is not a significant predictor of subjective performance measures such as software projects' process and product performance.[1] Since previous studies that used residual risk as a predictor of subjective performance provide incongruent results for firms in IT developed and IT developing countries, this research investigates whether the inconsistency persists when a taxonomy-based risk identification framework is used to predict subjective performance.

The primary purpose of this study is to develop and test a risk based model that predicts both objective and subjective performance in IT developing countries. Specifically, this study extends the work by Na and his colleagues by first assessing the impact of specific risk management strategies and residual risk on objective performance measures such as cost and schedule overruns. We then develop a taxonomy-based risk identification conceptualization of software development risk and assess its impact on both objective and subjective performance measures in an IT developing country. Thus, our study attempts to fuel the theory building process by providing descriptive results that should enhance our understanding of how software development risk and risk management practices impact objective and subjective software performance in an IT developing country.

We begin this study with a brief review of the relevant literature. We then define the measures and describe the models and resulting hypotheses. Next the methodology for collecting and analyzing the data is discussed. Finally, we report the results of the test of our models and discuss the implications of the study.

2. Background

2.1. Capability maturity model integration (CMMI)

Several researchers (c.f., Keil et al., 2000) have suggested that some software risk factors are consistent across organizations in different countries while other factors are country specific. To assist in determining the software development risk in an IT organization, the Software Engineering Institute at Carnegie Mellon University has developed a methodology that classifies IT organizations based on their level of software and engineering sophistication and management capability. The capability maturity model integration (CMMI) classifies software organizations into five levels based on the sophistication of their engineering and management practices. Organizations without systematic software engineering methods and tools are classified as level 1 or 2. This group's software development performance relies primarily on factors such as managerial experience and competence. Firms with process improvement capabilities that allow them to meet schedule, cost, quality and functionality targets are classified as 4 or 5 (Paulk, 2001; Paulk et al., 1995). The increase in IT outsourcing has led to an increased interest in the CMMI classification. In fact, these customer requirements have led to the development of *fast track* processes designed to facilitate rapid CMMI classification (Thomas and Smith, 2001). One of the conceptualizations of risk used in this study is derived from the CMMI framework. In addition, we use the CMMI framework to distinguish between firms in IT developed and IT developing countries.

2.2. Software development risk

Our study is based on two distinct conceptualizations of software development risk: *residual performance risk* (Nidumolu, 1995, 1996b) and *taxonomy-based risk identification* (Carr et al., 1993). Residual performance risk refers to the performance risk that remains after all the project planning and requirements analyses have taken place. Thus, residual performance risk is the intrinsic risk remaining during the latter stages of a software project (Nidumolu, 1996b). According to Na et al. (2004), there are two components of residual performance risk: residual controllable risk and unforeseeable risk. While the residual controllable risk continues through the latter stages, it can be mitigated in various ways through specific remedies. However, unforeseeable risk can be neither identified nor controlled.

[1] Process performance is a performance metric for the software development process and can be described by the (1) learning that occurs during the course of the project, (2) the degree to which management controls the project, and (3) the quality of the interactions between the IS team and users during the development process. Product performance is a performance metric that captures the performance of the finished product and can be described by the (1) technical performance of the software, (2) the degree to which the software conforms to user needs, and (3) the degree to which the software is flexible in supporting new products and changing user needs (Nidumolu, 1996b; Na et al., 2004). We discuss two categories of software development performance in the next section.

Risk management strategies are typically employed in the development process to reduce the risk inherent in software projects (Boehm, 1991; Chittister and Haimes, 1994, Chittister and Haimes, 1996; Fairley, 1994). For instance, Nidumolu (1996b) found that US software development firms reduced residual risk by standardizing the development process. He also found that increased levels of project requirements uncertainty are associated with higher residual risk. Na et al. (2004) replicated these results with data from Korean software development firms.

A popular alternative conceptualization of risk is the taxonomy-based risk identification framework which is derived from the capability maturity model integration (CMMI) framework and the Taxonomy-Based Questionnaire (TBQ). This method of assessing risk is designed to facilitate the systematic and repeatable identification of risks connected with the development of a software project (Carr et al., 1993). The TBQ is a standardized method of evaluating the level of organizational risk. While this taxonomy is simply a tool to organize development risks into various hierarchical levels, it is designed to extract the wide range of risks inherent in a software project. The TBQ assesses risk in three areas: project engineering, development environment, and program constraints. The TBQ organizes risks associated with these three areas into three levels: class, element, and attribute. TBQ has become much more common among software development organizations attempting to achieve higher CMMI levels (Williams et al., 2004).[2]

2.3. Performance

Software development performance can be grouped into two general categories: *subjective* performance and *objective* performance. Subjective performance assessment may be defined as an evaluation method that reflects the opinion of the people involved (Wohlin et al., 2000). While this technique has the advantage of easy data collection, it has difficulties with standardization since the project evaluation is dependent on the specific manager's judgment (Gray et al., 1999; Vallett and Condon, 1993). Two commonly used subjective performance constructs are *process performance* and *product performance*. These two subjective performance measures have been studied extensively in the software development literature (Nidumolu, 1996b; Wallace et al., 2004b; Rai and Al-Hindi, 2000).

In contrast, objective performance includes more quantifiable measures such as cost, effort, and schedule overrun (Gray et al., 1999). Because software development performance measures have different implications to different organizations, the extant literature often recommends using both subjective and objective performance measures (Briand et al., 1998; Gray et al., 1999). For example, compared to software development organizations, software order organizations are generally more concerned with quantifiable objective performance measures. Most existing research related to the performance of software development projects focuses on the development organizations rather than the order (or acquisition) organizations. Thus, most studies have tried to identify the factors that impact subjective performance measures (i.e., process and product performance). This performance classification scheme is depicted in Table 1.

Table 1
Project performance classification scheme

	Subjective performance	Objective performance
Measures	Process performance Product performance	Cost overrun Schedule overrun
Interested party	Development organization	Order/acquisition organization

Nidumolu's (1996b) study of US based software development firms found that residual risk is negatively associated with both process and product performance. Na et al.,'s replication of this study with software development firms in an IT developing country found that residual risk is not a significant predictor of subjective performance measures such as process and product performance. Moreover, these researchers found that the absolute level of residual risk in Korean firms is significantly higher than the residual risk reported by the US firms. Several explanations were offered for the disparate findings. One explanation was that the residual risk remaining during the latter stages of the projects in developing IT countries is so large that there is insufficient unique variation to capture the specific relationship between residual risk and broad subjective performance measures. By examining the role played by residual risk in influencing objective performance measures, our paper complements prior studies and yields new managerial insights for software development.

3. Measures

Five of the measures used in this study are drawn directly from studies by Nidumolu (1996b) and Na et al. (2004). Four new measures were developed specifically for this research. We use the same antecedent variables (i.e., standardization and requirement uncertainty) and mediating variable (i.e., residual risk), and subjective performance (process and product performance) used in the Nidumolu and Na et al. studies. The objective performance measures (i.e., cost and schedule overrun) and the taxonomy-based risk identification measure (i.e., functional development risk and systems development risk) were developed specifically for this study.

Requirement uncertainty reflects the absence of information needed by the development team to meet the user's requirements. The *standardization* variable represents only

[2] TBQ was originally proposed for the CMM framework in 1993. While CMM is being replaced by the CMMI, SEI's risk taxonomy is still widely used for software risk identification and evaluation. In fact, this taxonomy has recently been proposed by SEI to identify operational risks (Gallagher et al., 2005).

standardized control which consist of both output and behavioral control. Our measure includes output controls such as the creation of a system of milestones for the development team and behavioral controls such as tools and techniques prescribed by management. *Residual risk* represents the risk remaining during the later stages of the software project after the prescribed risk management strategies have been implemented. *Process performance* reflects the quality of development process. *Product performance* reflects the performance of the finished product.

A detailed description of the conceptualization of these constructs can be found in Nidumolu (1996b) and Na et al. (2004). Our objective performances consisted of an assessment of cost overrun and schedule overrun. For project *cost overrun*, firms were asked to indicate the approximate percentage, if any, the actual costs for the project exceeded the budgeted cost. For project *schedule overrun* firms were asked to indicate the approximate percentage, if any, the actual systems and programming effort for the project exceeded the budgeted completion time.

A popular alternative conceptualization of risk (i.e., taxonomy-based risk identification) derived from the SEI's Taxonomy-Based Questionnaire (TBQ) was developed for Models 2 and 3. This method of assessing risk is designed to facilitate the systematic and repeatable identification of risks connected with the development of a software project. While the TBQ is designed to assess risk related to product engineering, development environment, and program constraints, we consider only product engineering since this component covers the traditional software engineering activities and the technical factors related to the deliverable product itself. The only omission from the product engineering framework was engineering specialties since this component contains maintenance issues rather than the actual development process which is the focus of this study. The four remaining elements of project engineering can be grouped into either functional development or system development risk. Requirements, design, and code and unit testing are all associated with early stage or functional risk that reflects the actual development of the software functionality. Thus, the elements affecting *functional development risk* encompass an array of development processes, ranging from the early requirements analysis to actual coding of the software. The Integration and Test elements relate primarily to later stage or *system development risk*. System development risk denotes a crucial phase of the

Table 2
Response format and sample items

Composite variable	Number of items	Cronbach's alpha	Mean	Response anchors	Sample items
Standardization	12	0.91	3.40	To what extent did project follow procedures . . . Complete ad hoc—followed detailed procedures	Procedure used to control software change Tools or techniques for coding software
Requirement uncertainty	7	0.70	3.10	Strongly agree–strongly disagree	Requirements identified at the beginning of the project were quite different from those existing at the end of the project A lot of effort had to be spent in reconciling the various requirements of this software
Residual performance risk	5	0.82	3.06	How difficult to estimate in later stage Very easy–very difficult	Whether the software would meet user needs What would be the cost of operating the software
Functional development risk	11	0.76	2.68	Strongly agree–strongly disagree	Requirements are changing or yet to be determined Design specifications are in sufficient detail to write the code
Systems development risk	4	0.63	2.94	Strongly agree–strongly disagree	There will be sufficient hardware to do adequate integration and testing Test specifications are adequate to fully test the system
Product performance	11	0.84	3.29	Very poor–very good	Reliability of the software Range of outputs that can be generated Ease of use of the software
Process performance	9	0.83	3.33	Very poor–very good	Knowledge acquired by the firm about the use of key technologies Control of project cost Adherence to auditability and control standards
Cost overrun[a]					By approximately what percentage, if any, did the actual cost for the project overrun the originally budgeted costs?
Schedule overrun[a]					By approximately what percentage, if any, did the actual completion time for the project overrun the originally budgeted completion time?

[a] Cost and schedule overruns were single item measures, and were not subjected to the same analysis as the multi-item measures.

software development process where the product is actually executed and implemented into the system.

Since this study is an extension of the Nidumolu (1996b) and Na et al. (2004) studies, we utilized measure development and analysis similar to the process described in those two studies. Specifically, multi-item scale was subjected to item-to-total correlation analysis to identify potential item contaminants. Next, the factor structure of each scale was assessed through the application of principal component factor analysis. Items that impaired internal consistency or cross-loaded were eliminated. Each item was then weighted by factor coefficient to create the composite variable used in the study. Sample items, response anchors, number of items, means and reliability scores are reported in Table 2.

The reliability scores for the multi-item measure that were replicated from the Nidumolu and Na studies range from 0.70 to 0.91. These scores represent an acceptable level of internal consistency. The same process was followed to develop the new functional development risk and systems development risk measures. Items from the project engineering component of the TBQ survey were assigned to either functional risk or systems risk depending on whether the items were associated with the development phase or implementation phase of product development. Next, a confirmatory factor analysis was conducted to ensure that the items converged on the appropriate factor. Finally, the items were subjected to a reliability test. The reliability scores for the new functional and systems risk measures (0.63 and 0.76) reveal a marginally acceptable level of internal consistency. Since the objective performance measures (i.e., cost and schedule overrun) are single item measures, they were not subjected to the same measurement analyses.

4. Model development

Three distinct models are proposed in this study. First, we develop Model 1 which assesses the impact of two popular risk management strategies and residual risk on objective performance. Second, Model 2 assesses the impact of functional and systems development risk on objective performance. Finally, Model 3 assesses the impact of functional and systems development risk on subjective performance. The models are shown in Figs. 1–3. The hypothesized signs of path coefficients are shown in parentheses.

4.1. Description of Model 1

Model 1 (see Fig. 1) demonstrates the effects of standardization and requirement uncertainty on residual risk and the impact of residual risk on two objective measures of risk. To extend the theoretical hypotheses proposed in Nidumolu (1996b) and Na et al. (2004), we posit that there is a positive relationship between residual risk and cost and schedule overrun. In other words, an increase in residual risk will be associated with an increase in cost and schedule overruns.

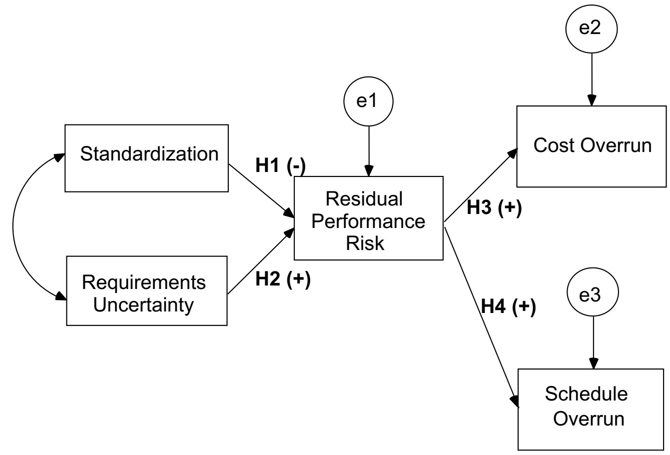

Fig. 1. Residual risk and objective performance measures (Model 1).

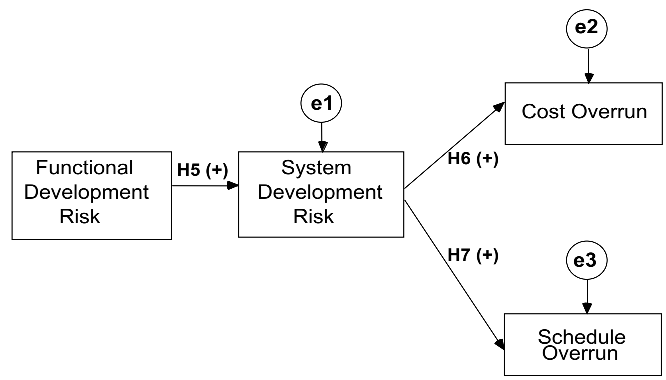

Fig. 2. Development risks and objective performance measures (Model 2).

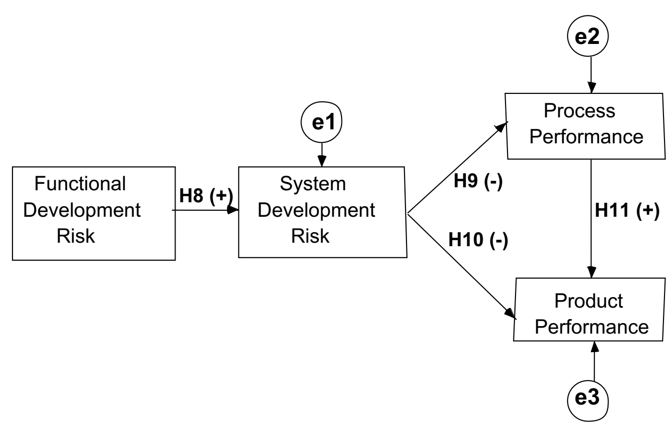

Fig. 3. Development risks and subjective performance measures (Model 3).

Within the CMMI framework, standardization is an important and complex construct. Standardization tools and procedures are often employed during the early stages of software development to control project delays and cost overruns (e.g., Nidumolu, 1996b). A highly standardized environment unifies the development staff by facilitating more effective communication and a more cohesive culture. Thus, greater standardization increases the likelihood of control and predictability of the software performance parameters. Because standardization throughout the project will inherently reduce the risk that remains during the later stages of product development, we offer the following hypothesis:

H1: An increase in standardization will be directly associated with a decrease in residual performance risk.

Requirements uncertainty is persuasive in the early stages of product development because of the inherent difficulties in understanding the complete requirements of a project (Nidumolu, 1995). Moreover, requirements uncertainty is more problematic in novel projects where tasks and processes are inherently more unstructured. Residual performance risk is intrinsically related to requirement uncertainty since unclear or ambiguous requirements that are open to frequent change make it difficult to reduce risk during the early stages of software development. Thus, we offer the following hypothesis:

H2: An increase in requirements uncertainty will be directly associated with an increase in residual performance risk.

While the extant literature reports a variety of measures of objective performance measures, cost and schedule overrun are the most common (Wallace and Keil, 2004). These two performance measures have the advantage of being quantitative, easy to measure, and logically important to software development projects. As the risk remaining during the later stages of software development increases, it is more likely that cost and schedule overruns will occur. Therefore, we hypothesize that residual performance risk will have a detrimental effect on software development cost and schedule.

H3: An increase in residual performance risk will be positively associated with cost overrun.
H4: An increase in residual performance risk will be positively associated with schedule overrun.

4.2. Description of Model 2

Model 2 (see Fig. 2) offers a more expansive conceptualization of software development risk that considers both early and late stage risk. Functional and system development risks are important components of a taxonomy-based risk identification framework (Carr et al., 1993). Both components are important in determining whether or not user requirements and customer expectations will be met. While functional development risk reflects the early stage risk related to specification requirements, design, and code and unit testing, system development risk reflects the risks associated with product integration and implementation. The two risk variables are closely related since the successful completion of early development stage tasks substantially impacts the outcome of latter stage system integration. Greater risk in the functional development stage should be related to greater risk in the system development stage. Hence, we propose the following hypothesis:

H5: An increase in functional development risk will be directly associated with an increase in system development risk.

The relationship between objective performance and risk becomes more apparent in the system integration stage later in the software development process. A well managed, low-risk software system integration and testing phase often results in a lower likelihood of schedule and budget overruns. Conversely, a project characterized by high levels of risk during the system development phase will be more likely to experience cost and schedule overruns (Ropponen and Lyytinen, 2000; Wallace and Keil, 2004). These observations are summarized in the following two hypotheses:

H6: An increase in system development risk will be positively associated with cost overrun.
H7: An increase in system development risk will be positively associated with schedule overrun.

4.3. Description of Model 3

Model 3 (see Fig. 3) is similar to Model 2 except Model 3 assesses the impact of the taxonomy-based risk identification measure on subjective performance. The relationship between functional development risk and system development risk described in H5 is repeated in the following hypothesis:

H8: Increase in functional development risk will be directly associated with increase in system development risk.

Managers are typically concerned about software project risk that cannot be effectively mitigated in early stages of software development (Na et al., 2004; Wallace et al., 2004a). We posit that the risk present in the system development phase should negatively impact subjective performance measures such as process and product performance. Several researcher (cf. Nidumolu, 1996b), have stressed the importance of evaluating both process and product performance since there is a potential for conflict between the efficiency of the process and product quality. Process performance is captured in the learning, control, and quality of interactions. Risk associated with system development can undermine these three dimensions which make controlling the process performance and product performance more difficult. The following hypothesis reflects the proposed relationship between systems risk and subjective performance.

H9: Increases in system development risk will be associated with decrease in process performance.

H10: Increases in system development risk will be associated with decrease in product performance.

Process and product performance are inherently related. While process performance reflects the IT staff's control over and understanding of the development phase of the project, product performance often depends on the level of process performance. Wallace et al. (2004a) noted that projects hampered with problems during its development process are unlikely to deliver successful software products. Thus we posit the following hypothesis:

H11: Increases in process performance will be associated with increase in product performance.

5. Methodology

Since the primary purpose of this study is to develop and test risk based models that predict both objective and subjective performance in IT developing countries, the data was collected from software development firms in Korea where the CMMI level is known to be much lower than in IT developed countries such as the US and India. Less than 10 South Korean firms participated in the 2001 SEI Software Capability Evaluation. According to the SEI (2005) Process Maturity Profile Mid Year Update, only 77 South Korean software firms participated in the 2005 SEI Software Capability Evaluation while 1982 US firms participated. Only three Korean firms scored higher than CMMI level 3. Even after adjusting for the difference in the size of each country's software industry, Korea is far behind the US in CMMI-based Software Process Improvement (SPI). Moreover, these finding are consistent with the Korean government report (Na, 1999) suggesting that most Korean firms are at CMMI level 1 or 2.

The data for this study was collected from three of the largest software firms in Korea with a minimum of 25,000 employees. These large scale organizations provide an appropriate setting for this study since they require extensive cooperation, communication and autonomy to successfully complete projects. For instance, most project teams included representatives from two-to-three external organizations. The data collection process was planned and the teams were selected in cooperation with the Software Quality Management Department Director for each firm. These firms were involved in 123 software development projects. Each project corresponds to one team. There was no overlapping among the teams. Each project leader was mailed a questionnaire that had been endorsed by their department director. All of the 123 project teams responded to the questionnaire. Each firm was equally represented with approximately thirty three projects each. A profile of the projects is presented in Table 3.

Table 3
Project profile statistics ($N = 123$ projects)

Project Attributes	Mean	Std. Dev	Minimum	Maximum
Project cost (in thousand dollars)	5838	12,794	4	47,145
Effort (person months)	36	46	2	200
Project duration (months)	21	19	2	120

6. Results

We test all 11 hypotheses in three structural equation models (SEM) utilizing AMOS 4. SEM is especially useful when research focuses on latent constructs (e.g., risk, uncertainty) which cannot be directly measured since the researcher can explicitly model the measurement error. SME is a multi-variate technique that combines the attributes of both factor analysis and multiple regression to simultaneously estimate a series of dependence relationships. SME is considered more powerful than multiple regression since it accommodates correlated independent variables, measurement error, correlated error terms and multiple independent and dependent variables each with multiple indicators (Hair et al., 1998). The parameter estimates and model fit statistics from this analysis are reported in Tables 4–6.

6.1. Results of Model 1

Consistent with Na et al. (2004), the results presented in Table 4 strongly support H1 and H2. There is a strong

Table 4
Model 1 results

Hypotheses	Paths	Standardized estimate	Significance
H1	Standardization → residual risk	−0.116	0.031
H2	Requirement uncertainty → residual risk	0.204	0.009
H3	Residual risk → cost overrun	0.222	0.032
H4	Residual risk → schedule overrun	0.210	0.042

X^2 (5 df) = 46.02, $p = .000$, NFI = .95, CFI 0.96, RMSEA = .26.

Table 5
Model 2 results

Hypotheses	Paths	Standardized estimate	Significance
H5	Functional development risk → systems development risk	0.309	0.007
H6	System development risk → cost overrun	0.198	0.032
H7	System development risk → schedule overrun	0.001	0.041

X^2 (3 df) = 47.76, $p = .000$, NFI = .94, CFI 0.94, RMSEA = .35.

Table 6
Model 3 results

Hypotheses	Paths	Standardized estimate	Significance
H8	Functional development risk → systems development risk	0.315	0.006
H9	System development risk → process performance	−0.337	0.000
H10	System development risk → product performance	−0.214	0.004
H11	Process performance risk → product performance risk	0.574	0.000

X^2 (2 df) = 0.989, $p = .61$, NFI = .99, CFI 0.99, RMSEA = .001.

negative relationship between standardization and residual performance risk, and a strong positive relationship between requirements uncertainty and residual performance risk. The results also support H3–H4 which posits that residual performance risk is positively related to both cost and schedule overruns. Hence, cost and schedule overruns increase when the residual risk remaining after risk management strategies have been employed increases.

6.2. Results of Model 2

The results in Table 5 provide strong support for H5–H7. First, functional development risk is related positively to systems development risk (H5). Therefore, early stage development risk significantly impacts later stage systems risk. Second, the results support the prediction that both objective performance measures (i.e., cost and schedule overruns) are related positively to system development risk (H6–H7).

6.3. Results of Model 3

The results in Table 6 strongly support H8–H11. Consistent with the results from Model 2, the relationship between functional and system development risk (H8) is strong and in the proposed direction. The results further indicate that system development risk negatively impacts the subjective performance measures (H9–H10). Hence, increased levels of systems development risk is negatively associated with both process and product performance. In addition, the results support our prediction of a significant positive relationship between process and product performance (H11).

In summary, the fit statistics presented at the bottom of Tables 4–6 suggest that, that the data is reasonably consistent with the hypothesized models. The X^2 values and significant levels (<0.05) for Model 1 and Model 2 indicate a lack of fit between the model and the data as might be predicted given the sample size (James et al. 1982). Yet, on other criteria the model performs very well (i.e., all GFI and CFI exceed the 0.90 threshold proposed by Bentler, 1990).

Table 7
Software development performance measure evaluation methods

Risk Type	IT developed country (US)	IT developing country (Korea)
Residual performance risk	Subjective, objective not tested yet	Only objective
Functional/system development risk	Neither tested yet	Both subjective and objective

Our results from the tests of the three models support all 11 hypotheses (H1–H11). All paths are significant and in the proposed direction. A representation of the results from our study and the two previous studies is illustrated in Table 7. Nidumolu (1996b) investigates how residual performance risk impacts subjective performance measures in the US. He did not consider any objective performance measures in his model. Na et al. (2004) tested Nidumolu's model with data from software development firms in Korea that were known to possess less developed IT capabilities than the US firms studied by Nidumolu. Na and his colleagues found no significant relationship between residual performance risk and the two subjective performance measures. Our study complements both of these studies by investigating the relationship between residual performance risk and objective performance measures. The current study extends this line of research by proposing two new risk constructs and testing their impacts on both objective and subjective performance in an IT developing country. The next logical extension of this research stream is to study the relationships between the *taxonomy-based risk identification* (Software Engineering Institute, 2005) construct developed in current study and objective and subjective performance measures in IT developed countries.

7. Discussion and conclusion

Risk is an inherent component of software development projects. Several studies have been conducted to help software managers to better understand both the drivers and resulting outcomes of software development risk. Our paper is motivated by two related studies (Nidumolu, 1996b; Na et al., 2004). Nidumolu (1996b) demonstrates that, in an IT developed country like the US, residual performance risk influences software process and project performance. He also reports that development standardization and user requirements analysis are two effective techniques for reducing residual performance risk. Na et al. (2004) test the same model with data from the IT developing country Korea. Their primary finding is that, unlike the results reported by Nidumolu (1996b), residual performance risk does not significantly impact subjective performance. Na and his colleagues suggest that the considerable difference between the two countries' average IT capability maturity levels may explain the inconsistent empirical findings. This argument is supported by the much higher levels of residual risk found in the Korean firms.

Our study addresses some important issues left unexplored in the two previous studies. Specifically, we investigate how different risk constructs impact both subjective and objective performance measures. Three conceptual models are developed and tested in our study. The results suggest that they are all robust tools for gauging objective and subjective performance. Our first model studies the impact of residual performance risk on two objective performance measures (cost overrun and schedule overrun). The literature identifies these two objective performance measures as being among the most important ones to IT managers and their bottom line. Hence, this model provides insights into managerial strategies to reduce the possibility of software project overruns. In the other two models, we developed *a taxonomy-based risk identification* conceptualization of risk that captured both early stage functional development risk and later stage system development risks. Our findings reveal that both functional and system development risks are important predictors of software project performance. Hence, this framework may be more promising than the residual risk framework in the assessment of risk factors that impact both objective and subjective performance.

While the empirical results from our study are compelling, there are limitations that should be considered. First, the data set used to test our models comes from only one IT developing country. In order to test the robustness our hypotheses across different countries, data should be collected in other countries. Second, our study does not include other constructs that may significantly affect software project performance. For example, managerial incentives, team member conflicts and information asymmetries could exacerbate software project overrun (Jiang and Klein, 2000; Smith and Keil, 2003; Keil et al., 2004). The omission of causal variables that are correlated with other variables in our structural model may impact the parameter estimates (e.g., Kline, 2004). Third, Xia and Lee (2005) reported that software project complexity plays an important role in systems development. We believe that including project complexity in our models could result in some additional insights into software project management. Finally, the assumptions inherent in our rationale for using CMM-I as the basis for distinguishing between IT developed and IT developing countries may not hold in all countries.

Our findings are relevant to both software development organizations and software order/acquisition organizations. With the current trend of outsource software development to foreign countries such as Korea, our study highlights the importance and relevance for understanding software development risk and project performance in foreign countries. It is often suggested that outsourced software projects usually exhibited significantly higher levels of management and control risk (Wallace et al., 2004b). To address this important software management issue, our study provides evidence from Korea software industry that reveals which risk measures are relevant in predicting certain performance measures. With this information software managers can more accurately pinpoint the cause of certain inefficiencies in software development process. These insights can lead to a more efficient allocation of their resource in software risk management. For example, our study suggests that software outsourcing partners share common interests in controlling functional development risk and system development risk in software development. While a previous empirical study implies that Korean software managers may lack strong incentives to control residual risk when they are concerned primarily with subjective performance, this study demonstrates that residual risk reduction matters to both outsourcing partners when objective performance measures are emphasized in their bilateral relationship. How to properly align the two outsourcing parties' incentives to better manage software development risk is, in our opinion, an interesting research question that merits further theoretical and empirical investigations.

Acknowledgements

This research was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-042-B00057) and the University of Alabama in Huntsville Center for the Management of Science and Technology (CMOST).

The authors would like to thank Ms. Annakena Plaza and Ms. Shannon Polkowski for their assistance in the preparation of the manuscript.

References

Barros, M.O. et al., 2004. Supporting risks in software project management. Journal of Systems and Software 70, 21–35.

Bentler, P.M., 1990. Comparative fit indexes in structural models. Psychological Bulletin 107, 238–246.

Boehm, B., 1991. Software risk management: principles and practices. IEEE Software 8 (1), 32–41.

Briand, L.C., Emam, K.E., Bomarius, F., 1998. COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment. In: Proceedings IEEE International Conference on Software Engineering, pp. 390–399.

Carr, Marvin, Konda, Suresh, Monarch, Ira, Ulrich, Carol, Walker, Clay, 1993. Taxonomy based risk identification (CMU/SEI-93-TR-6, ADA266992), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Charette, R., 2005. Why software fails? IEEE Spectrum, 42–49.

Chittister, C., Haimes, Y., 1994. Assessment and management of software technical risk. IEEE Transactions on Systems, Man, and Cybernetics 24 (2), 197–202.

Chittister, C., Haimes, Y., 1996. System integration via software risk management. IEEE Transactions on Systems, Man, and Cybernetics 26 (5), 521–532.

Deloitte Consulting, 2005. Calling a change in the outsourcing market: the realities for the world's largest organizations, April 2005, Deloitte Consulting LLP.

Fairley, R., 1994. Risk management for software projects. IEEE Software 11 (3), 57–67.

Gallagher, B. et al., 2005. A taxonomy of operational risks. Technical Note, CMU/SEI-2005-TN-036, September 2005.

Goldstein, H., 2005. Who killed the virtual case file? IEEE Spectrum, 24–35.

Gray, A.R., MacDonell, S.G., Shepperd, M.J., 1999. Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgment. In: Sixth IEEE International Symposium on Software Metrics, pp. 216–227.

Hair, Joesph F., Tatham, Rodnald, Anderson, Rolph, Black, William, 1998. Multivariate Data Analysis, fifth ed. Prentice-Hall.

James, L.R., Mulaik, S.A., Brett, J.M., 1982. Causal Analysis: Assumptions, Models, and Data. Beverly Hills: Sage Publications, Inc.

Jiang, J., Klein, G., 2000. Software development risks to project effectiveness. Journal of Systems and Software 52, 3–10.

Keil, M., Tan, B.C., Wei, K.K., Saarinen, T., Tuunainen, V., 2000. A cross-cultural study on escalation of commitment behavior in software projects. MIS Quarterly 24 (2), 299–325.

Keil, M., Smith, J., Pawlowski, S., Jin, L., 2004. Why didn't somebody tell me? Climate, information asymmetry, and bad news about troubled projects. DATA BASE for advance in Information Systems 35 (2), 65–84.

Kline, R.B., 2004. Principles and practice of structural equation modeling, second ed. Guilford Press, New York.

Minevich, M., Richter, F., 2005. Global Sourcing Report 2005. CIO Insight Magazine, March 2005.

Na, K., 1999. Software quality policy for improving competitiveness in Korea software firms. Government Report, Korea Ministry of Information and Communication.

Na, K., Li, X., Simpson, J.T., Kim, K., 2004. Uncertainty profile and software project performance: a cross-national comparison. Journal of Systems and Software 70, 155–163.

Nidumolu, S., 1995. The effect of coordination and uncertainty on software project performance: residual risk as an international variable. Information System Research 6 (3), 191–219.

Nidumolu, S., 1996a. A comparison of the structural contingency and risk-based perspectives on coordination in software-development projects. Journal of Management Information Systems 13 (2), 77–113.

Nidumolu, S., 1996b. Standardization, requirements uncertainty, and software project performance. Information and Management 31, 131–150.

Paulk, M.C., 2001. Extreme programming from a CMM perspective. IEEE Software 18 (6), 19–26.

Paulk, M.C., Curtis, B., Chrissis, M.B., Weber, C.V., 1995. The Capability Maturity Model for Software. Addison-Wesley Publishing.

Rai, A., Al-Hindi, H., 2000. The effects of development process modeling and task uncertainty on development quality performance. Information and Management 37, 335–346.

Ropponen, J., Lyytinen, K., 2000. Components of software development risk: how to address them? A project manager survey. IEEE Transactions on Software Engineering 26 (2), 98–112.

SEI, 2002. Process maturity profile of the Software Community 2001 year end update, Software Engineering Institute, Carnegie Mellon University.

SEI, 2005. Process maturity profile: Software CMM 2005 mid-year update, Software Engineering Institute, Carnegie Mellon University.

Smith, J., Keil, M., 2003. The reluctance to report bad news on troubled software projects: a theoretical model. Information Systems Journal 13 (1), 69–95.

Software Engineering Institute, 2005. Introduction to the CMMI® Acquisition Module (CMMI-AM)—Module 2: CMMI-AM and Project Management, Software Engineering Institute, Carnegie Mellon University, Pittsburg, PA.

Thomas, G., Smith, H., 2001. Using structured benchmarking to fast-track CMM process improvement. IEEE Software 18 (5), 48–52.

Tiwana, A., Keil, M., 2004. The one-minute risk assessment tool. Communications of the ACM 47 (11), 73–77.

Vallett, J., Condon, S.E., 1993. The (Mis)use of subjective process measures in software engineering. In: Proceedings of the Eighteenth Annual Software Engineering Workshop: Software Engineering Laboratory Series, SEL-93-003, pp. 161–168.

Wallace, L., Keil, M., 2004. Software project risks and their effect on outcomes. Communications of the ACM 47 (4), 68–73.

Wallace, L., Keil, M., Rai, A., 2004a. How software project risk affects project performance: an investigation of the dimensions of risk and an exploratory analysis. Decision Science 35 (2), 289–321.

Wallace, L., Keil, M., Rai, A., 2004b. Understanding software project risk: a cluster analysis. Information and Management 42, 115–125.

Weakland, T., 2005. DiamondCluster 2005 Global IT Outsourcing Study, Spring of 2005.

Williams, R., Ambrose, K., Bentrem, L., 2004. A roadmap of risk diagnostic methods: developing an integrated view of risk identification and analysis techniques. Technical Note, CMU/SEI-2004-TN-002.

Wohlin, C., Mayrhauser, A.V., Host, M., Regnell, B., 2000. Subjective evaluation as a tool for learning from software project success. Information and Software Technology 42, 983–992.

Xia, W., Lee, G., 2005. Complexity of information systems development project: conceptualization and measurement development. Journal of Management Information Systems 22 (1), 45–83.

Kwan-Sik Na is an Associate Professor in the Business at the Seowon University in Korea and affiliate faculty member in the Graduate School at the University of Alabama in Huntsville. He received his Ph.D. in Management Information Systems and Management Science at the Kwangwoon University in Korea. His research focuses on Software and Information System Risk Management, Decision Support System and Software Industry Research. He has published over 30 articles including *Journal of Systems and Software*.

James T. Simpson is Professor of Marketing and Chairman of the Department of Management and Marketing at the University of Alabama in Huntsville. He received his Ph.D. in Marketing and Applied Statistics at the University of Alabama. His research focuses on Technology Management, Marketing High Technology Product and Services, and Marketing Channel Structure and Behavior. His research has appeared in numerous journals including the *Journal of Marketing Research, Marketing Letters, Journal of Business Research, Journal of Systems and Software* and*European Journal of Innovation Management.* He has lectured and served as a visiting scholar at universities in Russia, Byelorussia, China, Romania, Taiwan, Ireland and England.

Xiaotong Li is an assistant professor of MIS at the University of Alabama in Huntsville. He received his Ph.D. in MIS from the University of Mississippi. His research has appeared in many journals including *Communications of the ACM*, *Marketing Science*, *IEEE Transactions*, *Journal of Systems and Software* and *Information Resource Management Journal*. His major research interests are in Game Theory and Economics of IT. He has been invited to give research seminars at University of Minnesota (MIS Research Center), Rensselaer Polytechnic Institute, Hong Kong Polytechnic University and other major universities. He recently won the best paper award from *IEEE Transactions on Engineering Management*.

Tushar Singh is a Business Development Specialist for Tefen USA, operations consulting firm based in New York City. He received his undergraduate degree in Electrical Engineering from the Georgia Institute of Technology and Masters of Science in Management from the University of Alabama in Huntsville. His research has been published in *Optik and Journal of Modern Optics*.

Ki-Yoon Kim is Professor of Business Administration, Chairman of the Department of Business Administration at the Kwangwoon University in Korea. He received his Ph.D. in Management Science at the Korea University in Korea. His research focuses on IT Risk Management, and Software and Information System Risk Management. He has published over 50 articles in these areas including *Journal of Systems and Software* and *Journal of Information System Education*.