



Requirement Volatility, Standardization and Knowledge Integration in Software Projects: An Empirical Analysis on Outsourced IS Development Projects

Rajesri Govindaraju*, Angra Bramagara, Leksananto Gondodiwiryo & Tota Simatupang

Industrial Systems and Techno-economics Research Group
Industrial Technology Faculty, Institut Teknologi Bandung
Jalan Ganesa 10, Bandung 40132, Indonesia

*Email: rajesri_g@ti.itb.ac.id

Abstract. Information systems development (ISD) projects are highly complex, with different groups of people having to collaborate and exchange their knowledge. Considering the intensity of knowledge exchange that takes place in outsourced ISD projects, in this study a conceptual model was developed, aiming to examine the influence of four antecedents, i.e. standardization, requirement volatility, internal integration, and external integration, on two dependent variables, i.e. process performance and product performance. Data were collected from 46 software companies in four big cities in Indonesia. The collected data were examined to verify the proposed theoretical model using the partial least square structural equation modeling (PLS-SEM) technique. The results show that process performance is significantly influenced by internal integration and standardization, while product performance is significantly influenced by external integration and requirement volatility. This study contributes to a better understanding of how knowledge integration can be managed in outsourced ISD projects in view of increasing their success.

Keywords: *external integration; internal integration; outsourced software projects; process performance; product performance; requirement volatility; standardization.*

1 Introduction

Software development is a highly complex task in which different groups of people need to collaborate and exchange their knowledge. This process may involve a high degree of uncertainty, which may result in high failure rates. In response to this problem, different strategies and techniques have been proposed to minimize the negative effects of the risks embedded in software projects. Strategies such as managing standardization and implementing requirement engineering have been used frequently in software engineering [1].

Several researches in information systems development (ISD) have been done to find factors that influence ISD project performance. A number of

investigations were focused on examining the effect of technical methods/techniques used in the software engineering process on ISD project performance. Studies that focused on examining the role of technical methods are among others, Nidumolu [2], Nidumolu [3], Zowghi and Nurmaliani [4], Na [5], Na, *et al.* [6], and Na, *et al.* [7]. These investigations mostly examined the influence of factors such as standardization, requirement uncertainty, inherent uncertainty, coordination, and requirement engineering practices. They looked at the influence of those factors on product performance and process performance.

Though organizations have applied a variety of methods and tools to address software development problems, approximately 75% of software development projects are delayed, over budget, or do not deliver core functionality [8]. Studies such as Barki, *et al.* [9], Wallace, *et al.* [10], Wang and Liu [11], Jiang, *et al.* [12], and Jun, *et al.* [13] have addressed the effect of management practices in their investigations.

While past researches were more focused on analyzing the roles of technical methods in increasing ISD project performance [14] and studying the effect of management practices on ISD project performance, recent research has suggested that ISD should be viewed more from a knowledge management perspective [14]-[17]. Moreover, most of the past studies focused on in-house development projects. Recently, however, companies have been increasingly outsourcing all or part of their activities to external vendors [18], including ISD activities. In outsourced ISD projects, client and vendor share the responsibility of managing the project. As the vendor faces a considerable amount of risk in an outsourced ISD project, an investigation into how to manage ISD project risk and performance from the vendor's perspective is important [19].

2 Literature Review

Various perspectives have been used to examine why some organizations are more successful than others in executing ISD processes. Earlier studies on ISD were more focused on the application of technical principles to achieve a more manageable and predictable systems development process and performance outcomes [14]. Some of those studies were focused on managing risk in software projects by analyzing different aspects of the development process that create uncertainty in the process and how various techniques and methods can be applied to manage that uncertainty [10].

From this stream of studies, among others, Nidomolu [3] and Na, *et al.* [6] examined the effect of requirement uncertainty and standardization on process performance and product performance in software development projects.

Nidumolu [3] described the uncertainty of requirements as the difference between the information needed to identify user needs and the information possessed by the developers. The study identified three dimensions of uncertainty: requirement instability (which reflects the extent of changes in user requirements in the entire project), requirement diversity (which reflects the degree of difference among users about requirements), and requirement analyzability (the extent to which the process of producing requirement specification can be reduced to objective procedures).

Other studies proposed the concept of requirement volatility as an important characteristic of the software development lifecycle. Stark, *et al.* [20] defined requirement volatility as the change in the requirements that happens after the basic set of requirements has been agreed upon by both clients and developers or analysts. Requirement volatility is defined as “the number of requirement changes (addition, deletion, modification) in a specified time interval (week, month, year or in particular phase)” [21]. Hence, requirement volatility refers to growth or changes in requirements during a project’s development lifecycle [21]. Requirement volatility is often associated with the following terms: requirement change, requirement creep, scope creep, and requirement instability.

Standardization refers to the use of methodologies, tools, and techniques at a certain level of a particular project [3]. The application of standardization is usually set by superiors to be applied by subordinates [3]. Nidumolu [3] categorizes standardization as output control standardization and behavior control standardization. In software engineering research, output controls are often implemented by defining specific outcomes such as milestones at different phases of the ISD process [22]. Behavior controls are often implemented by a standardized definition of how individual software development tasks should be performed [23].

Wallace, *et al.* [10] argued that uncertainty in the performance of a software development project can be attributed to requirement related factors, development team related factors, and organization environment related factors. In line with this, research done by Jun, *et al.* [13] complements the studies conducted by Nidumolu [3] and Na, *et al.* [6] to include several factors associated with the development team and organizational environment, and assesses the influence of these factors on process performance and product performance in software development projects. Factors studied were the project’s inherent uncertainty, internal integration, user participation, and project planning and control. Participation refers to the condition where one of the parties participates when the other party contributes to something [9]. User participation consists of all activities to improve communication and exchange

of information and to share the understanding of concepts through personal coordination [9]. User participation is related to the concept of horizontal coordination between users and developers that can bridge the gap between developers and users [9].

Recently, the importance of knowledge integration for technology development projects in general was highlighted by among others, Purvis, *et al.* [15] and Carlile [24]. Furthermore, Patnayakuni, *et al.* [14], Patnayakuni, *et al.* [16] and Parolia, *et al.* [17] argued that the systems development process and its outcomes should be examined from a knowledge management perspective. Developing a customized software application requires deep knowledge about the business problem domain as well as technical knowledge about software development methodologies and technologies [25].

3 Research Model and Hypotheses

In this study, a conceptual model was developed, adopting the ideas from earlier models developed by Nidumolu [3], Barki, *et al.* [9] and Jun, *et al.* [13]. In Nidumolu [3] and Na, *et al.* [7], the influence of requirement uncertainty and standardization on process performance and product performance has been tested. In this study, the standardization construct was adopted and the requirement uncertainty construct was adapted into requirement volatility. The impact of the extent to which changes to requirements happen after the requirements have been agreed upon by both client and vendor on project performance was investigated. This way, the investigation performed in this study is slightly different from the investigation done by Nidumolu [3] and Na, *et al.* [6], who analyzed the impact of requirement uncertainty, i.e. the difference between the information needed to identify user needs and the information possessed by the developers, on the performance of ISD projects.

Furthermore, taking into account the importance of knowledge management in ISD projects, the issue of knowledge integration in projects was examined in this study. Knowledge integration needs to be managed internally and externally [25]. From the vendor's perspective, internal integration is mainly related to the firm's ability to put together and exploit all knowledge available inside the project team and to reduce the differences in understanding and expectation among the team members. On the other hand, external knowledge integration refers to the ability of the project team to absorb insights and knowledge about the business needs and existing technological environments in the client company, during the development process. Thus, two antecedents were included in the model, i.e. internal knowledge integration (briefly referred to as internal integration) and external knowledge integration (briefly referred to as external integration).

In Jun, *et al.* [13], the influence of internal integration on process performance and product performance has been tested and the study found that internal integration significantly influences both process performance and product performance. In the context of Indonesian vendors, these relationships were revalidated in the present study. Related to external integration, previous studies highlighted the importance of user participation in software projects to provide system requirements (i.e. Barki, *et al.* [9], Wang and Liu [11], Jun, *et al.* [13]). However, the concept of external integration used in this study refers to the role of the client company in providing knowledge, which is broader than user participation used in earlier studies (e.g. Jun, *et al.* [13]). Various groups of people (actors) in the client company need to participate in order to provide all the support needed to deliver the expected project results. The actors include users (key users), IT specialists, and senior managers. Based on the review of literature on ISD research conducted for this study, very few or no studies have examined the contribution from different actors in the client company, including key users, senior managers and IT specialists, to integrating knowledge needed in ISD projects and its effect on ISD project performance.

In the developed model, ISD project performance is defined for two aspects, i.e. product performance and process performance (Nidumolu [3], Wallace *et al.* [10], Patnayakuni, *et al.* [14], Patnayakuni, *et al.* [16], Jun, *et al.* [13]). Product performance refers to the quality of the product or system that is delivered by the ISD team and process performance refers to the extent to which a project is delivered on schedule and within budget. Product performance measures the performance of the products that have been completed [6]. Product performance may be described by three components: the technical performance of the product, the level of comfort of the software as required by the users, as well as a degree of flexibility in supporting new user requirements [5]. In this study, product performance and process performance are defined as dependent variables.

3.1 Internal Integration

Technology development projects require the integration of various knowledge from multiple domains of expertise [25]. Internal knowledge integration refers to the extent to which a project team distributes and shares knowledge while completing different tasks in a software development project. Internal integration allows the blending of diverse technological and functional expertise in the process of developing the targeted solution in a software development project. Since the present study was executed from the vendor's perspective, internal integration is meant to address the integration of knowledge among the vendor's internal members, consisting of project manager, analysts (IS specialists), and developers.

Software development projects are knowledge intensive projects and therefore collaboration between team members with diverse skills and specializations is necessary [13]. Poor internal integration among team members can lead to conflicting interpretations of system requirements among the team members, while a higher level of internal integration allows various project members to develop a convergent understanding regarding the project goals, scope, and constraints [26]. This increase in convergent understanding helps team members to anticipate downstream development problems and to quickly solve and correct them [27],[28]. Besides, increased internal integration also stimulates team creativity, improves understanding of the project, helps to quickly fix problems that may arise during the development process, and helps reaching the targeted solutions [29].

Internal integration can improve coordination and collaboration between teams of developers [13]. In addition, it can prevent problems that could lead to a conflict among members of a development team. Yetton, *et al.* [30] found that conflicts within a project team can lead to instability of the team and thus result in project delay and cost overrun. In other words, a conflict within the team has a negative effect on project performance, in particular process performance. Based on the arguments presented above, the following hypotheses are proposed:

H1: Internal integration has a positive effect on process performance

H2: Internal integration has a positive effect on product performance

3.2 External integration

In an ISD project, various groups of people (actors), including business users, IT specialists, systems developers, and even senior managers, need to be involved to provide all the necessary knowledge and support needed in a project (SIAPA). In an outsourced ISD project, the vendor is responsible for delivering the information systems application. However, to develop the systems needed by the client company, it is important for the project team to get sufficient business and technological knowledge from the client side.

External knowledge integration is defined as the extent to which the client company shares the necessary knowledge with the project team, which includes business related knowledge and technology related knowledge. External integration can be seen as an indicator of how extensively the actors from the client company play their role as sources of business and technological knowledge, across all stages of the development process. External knowledge integration includes the integration of knowledge about business directions and

needs, regulatory constraints, as well as current and projected (future) technological environments that might affect an ISD project.

Senior managers, IT specialists and business departments' representatives may have a unique view of the factors that should be considered in defining an effective solution. External integration allows the blending of this diverse functional and technological expertise in formulating the technological solution. Thus, from an ISD point of view, external integration directs the attention toward the system's functionalities and constraints that are appropriate to deriving business benefits and providing an appropriate new IT environment.

Poor external integration among team members can lead to conflicting interpretations of project goals, overlooked requirements, and design decisions that are incompatible with the project's objectives. Such fragmentation of efforts and goals among project constituents is a leading cause of failure in technology development projects [31]. When different perspectives are integratively brought to bear within a project, it is more likely that the resulting solution will embody the organization's needs [32]. Research on ISs that are based on participative decision making (PDM) and planned organizational change (POC) theories indicates that user participation can increase the likelihood of successful implementation of information systems [2],[13],[33]. In addition, research done by Wang and Liu [11] showed a positive effect of user participation on product performance.

External integration also addresses the participation provided by the stakeholders from the client company to share knowledge for the purpose of evaluating the project results (deliverables). Knowledge derived from this evaluation is valuable input for the project managers to control the project and prevent problems in the downstream phases of the system's life cycle. However, Yetton [30] argued that user participation has two different kinds of effects on the variation of the project costs. User participation can increase the costs due to suggestions to change the specifications, but can also reduce the costs through managing expectations and resolving potential problems quickly. In summary, external integration describes the extent to which the client company participates in sharing knowledge through continuous communication with the vendor to ensure that the system is developed in accordance with the desired requirements and the project can be completed as planned. Based on the arguments presented above, the following hypotheses are proposed:

H3: External integration has a positive effect on process performance

H4: External integration has a positive effect on product performance

3.3 Standardization

The implementation of standardization is viewed as one of many solutions to the problems faced during the software development process [3]. The Software Engineering Institute's (SEI) capability maturity model highlights the importance for the organizations to define a standard in the software development process and describes an integrated set of engineering and management processes [34]. Based on Nidumolu [3] there are two types of standardization addressed: output control standardization and behavior control standardization. Software engineers and IS researchers have identified a number of ways in which standards can help reduce performance risks in software development [3]. The definition of different phases in an ISD project and the use of milestones in defining outputs of different stages of a development process have been used a lot in standardizing software development outputs [3],[22],[35]. Behavior controls are often implemented by a standardized definition of how software development tasks should be performed [23].

Standards permit large groups of developers to coordinate their activities more easily, thus reducing the likelihood of project delays and cost overruns [3]. They promote better communication among the participants in a project and between the project team and the managers they report to. Nidumolu [3] found that an organization-wide standard for output controls, such as the phases to be observed during the project, the milestones to be completed at each phase, the documents to be prepared at milestone completion, and the approval procedures to be followed at each milestone, have an important role in reducing performance risk and improving process and product performance. However, the organization-wide standardization of the tools and techniques to be used in developing software did not significantly decrease performance risk nor improve process and product performance.

Project environments contain many diversities, such as the diversity of values and culture brought by each personnel, diversity in the way personnel work, and others. Through standardization, all personnel involved are required to follow the standards set by the supervisor or project leader. With standardization, team members are "forced" to follow certain procedures that have been planned considering the budget and time available for the project. Apart from that, standardization facilitates the analysis needed in case there are problems related to the software development, in order to determine the cause of the problems so that the developers can solve them quickly.

Other studies highlighted the importance of standardization to increase project performance. Suh and Jeong [36] formulated the procedure regarding the development and standardization of technical requirements and they found that

standardization improves process performance. Na [5] studied the impact of standardization and requirement uncertainty on software project performance and found that standardization of software development reduces residual risk and therefore increases project performance. A standardized project environment facilitates more effective communication and a more cohesive culture, which can unite the team of developers [7]. Standardization in all parts of the project will inherently reduce the risk of failure of the project at a later stage of application development. Thus, increased standardization will improve the performance of control in software development projects. Na, *et al.* [7] proved that an increase in standardization reduces the risks related to project performance. Liu, *et al.* [37] found that software process standardization improves project performance. Based on the arguments presented above, the followings hypotheses are proposed:

H5: Standardization has a positive effect on process performance

H6: Standardization has a positive effect on product performance

3.4 Requirement Volatility

The success of any software project is directly related to the quality of its requirements [38]. In practice, system requirements often change during the development process, causing part of the software components to be modified, deleted or added. Requirements changing during the system development is known as requirement volatility [4],[38].

The change in requirements during the development life cycle has an impact on project costs, the project schedule and the quality of the product resulting from the project [38]. Requirement volatility can occur at different points during the software development process. Nidumolu [2] identifies three dimensions of uncertainty in the requirement identification process: requirement instability, requirement diversity, and requirement analyzability. Recent studies on requirement volatility found that among the various dimensions of requirement volatility (i.e. requirement instability, requirement diversity, and requirement analyzability, changes in business environment), only two dimensions were consistent in the analysis, i.e. requirement instability and requirement diversity. In this study, requirement volatility refers to the degree of instability and diversity of the system's requirements, captured during the project stage of an ISD project.

Among the studies that analyzed the impact of requirement volatility on software project performance, Stark, *et al.* [20] examined the impacts of requirement volatility on the project schedule and costs. Lane and Cavaye [39] investigated the impact of requirement volatility on the effectiveness and

efficiency of the software development process and found that there was no direct impact of requirement volatility on either of these two concepts. Previous works by Zowghi and Nurmuliani [40] and Zowghi, *et al.* [41] found that there was no strong evidence for an influence of requirement volatility on software development productivity. Furthermore, Wang and Liu [11] found that requirement uncertainty has an impact on the performance of a project, through increased performance risks. Na, *et al.* [7] showed that an increase in requirement uncertainty may increase risks related to the performance of the project. Considering the inconclusive empirical results found in previous studies, it is important to further investigate the impact of requirement volatility on the success of ISD projects. Based on the arguments presented above, the following hypotheses are proposed:

H7: Requirement volatility has a negative effect on process performance

H8: Requirement volatility has a negative effect on product performance

4 Research Methodology

4.1 Data Collection

The unit of analysis used in this study is an ISD project. In order to ensure the validity of the collected data, the following criteria were set for selecting the project samples: (1) the project involves at least 5 members, (2) the project team members must include system analysts, system designers, and programmers, (3) the project must be finished not more than one year ago. The participants filling in the questionnaire were ISD project managers or project team leaders, working in Indonesian software companies providing outsourced SD services. In order to prevent any confusion, the participants were asked to select their latest project as the sample for our survey. Thus, the situation in their last project was used as the basis for answering the questionnaire.

Table 1 Demographic characteristics of the samples.

	Profile	Frequency	Percentage
Position of the respondent	Project Manager	36	78.3%
	Project Team Leader	10	21.7%
Respondent's work experience (years)	3-5	15	32.6%
	6-10	15	32.6%
	11-15	10	21.7%
	>15	6	13.1%
Planned project duration (months)	4-6	7	15.2%
	7-12	35	76.1%
	13-18	4	8.7%
Number of project team members	5-10	27	58.7%
	11-20	19	41.3%

Data collection was done in four months, from December 2012 until March 2013. Questionnaires were distributed to 120 companies in four big cities in Indonesia: Jakarta, Bandung, Jogjakarta, and Surabaya. In total, 49 questionnaires were returned. However, only 46 questionnaires were usable. Three questionnaires were not used because they were incomplete. The demographic characteristics of the samples are presented in Table 1.

4.2 Measures

The research model consists of six variables of which 4 are independent variables and 2 others are dependent variables. Previous studies were used as a basis for the development of the operationalized measures used in this study. The operationalization of each variable is presented in Table 2.

Table 2 Operationalization of variables.

Variable	Indicator	Code	References
Internal Integration (II)	The project team meets frequently	II1	[9]
	Project team members are kept informed about major decisions concerning the project	II2	
	Every effort is made to minimize team turnover	II3	
	Project members actively inform other members regarding the progress they have made	II4	
External Integration (EI)	Key users from the client company actively provide knowledge needed in requirement definition	EI1	[13]
	Key users from the client company actively involved in discussing project deliverables and make contributions	EI2	
	Senior managers provide direction and regulation	EI3	
	IT specialists from client company provide information related to infrastructure and IT systems	EI4	
Standardization (ST)	The use of tools or techniques for project management	ST1	[3]
	The use of tools or techniques for generating requirements	ST2	
	The use of standards for documentations	ST3	
	The use of tools or techniques for system design	ST4	
	The use of tools or techniques for coding software	ST5	
	The use of tools or techniques for testing software	ST6	
	The use of tools or techniques for data administration	ST7	
Requirement Volatility (RV)	It was difficult for stakeholders to reach agreement among themselves on requirements	RV1	[3][4]
	A lot of effort had to be spent in incorporating the requirements of various users	RV2	
	Requirements fluctuated in the later stages	RV3	
	Differences in requirements were identified at the start of the project from the final requirements	RV4	

Variable	Indicator	Code	References
Process Performance (PC)	The project was completed within budget	PC1	[10]
	The project was completed within schedule	PC2	
Product Performance (PD)	The application developed is reliable	PD1	[10][42]
	The application developed is easy to use	PD2	
	The system meets user’s intended functional requirements	PD3	
	Users are satisfied with the system delivered	PD4	

Each questionnaire item was measured using a 5-point Likert-type scale ranging from “strongly disagree” (1) to “strongly agree” (5).

5 Analysis and Results

5.1 Measurement Model Evaluation

The partial least squares structural equation modeling (PLS-SEM) technique was used to examine the data, supported by the SmartPLS software. SEM is a collection of statistical techniques that allow a set of relations between one or more independent variables and one or more dependent variables to be examined, and it is referred to as causal analysis, simultaneous equation modeling, path analysis, or confirmatory factor analysis [43].

The measurement model was first examined to assess the reliability and validity of the constructs using confirmatory factor analysis (CFA) before the structural model was tested [44]. This measurement model evaluation was meant to analyze how the measured variables represent a construct that is not measured directly [44]. The convergent validity and discriminant validity of the model were evaluated. The convergent validity was assessed based on three indicators: factor loading, average variance extracted (AVE), and composite reliability (CR). The average variance extracted (AVE) is the average percentage of variation explained by the items in a construct. Composite reliability is defined as an approach of overall reliability and estimates consistency within the construct itself, including the stability and equivalence of the construct [44]. AVE and CR can be calculated using Eq. (1) and Eq. (2) respectively.

$$AVE = \sum K^2 / n \tag{1}$$

$$CR = (\sum K)^2 / [\sum K^2 + (\sum 1 - K^2)] \tag{2}$$

K = Factor loading for every item
 N = Number of items in a construct

Constructs have convergent validity when the factor loading of their indicators is above 0.6, the AVE is above 0.50, and CR is above 0.70 [44],[45],[46]. Based

on the results of the confirmatory factor analysis it was found that three indicators were not valid because of having factor loading values below 0.5. The invalid indicators were EI4, RV2, and ST3. Table 3 shows the values of factor loading, AVE and CR for each construct after the removal of the three invalid indicators.

To assess the discriminant validity, the square root of the AVE for each construct was compared with the inter-factor correlations between that construct and all the other constructs. If the square root of the AVE for each construct is higher than all the correlations between the construct and the other constructs, then the discriminant validity is good [44],[46]. As shown in Table 4, the discriminant validity was assured for all the constructs.

Table 3 Factor loading, average variance extracted and composite reliability.

Construct	Indicator	Factor Loading	AVE	CR
Internal Integration (II)	II1	0.748	0.6079	0.861
	II2	0.783		
	II3	0.721		
	II4	0.860		
External Integration (EI)	EI1	0.842	0.7066	0.877
	EI2	0.933		
	EI3	0.736		
Standardization (ST)	ST1	0.657	0.5326	0.872
	ST2	0.709		
	ST4	0.769		
	ST5	0.714		
	ST6	0.841		
Requirement volatility (RV)	RV1	0.546	0.5245	0.759
	RV3	0.645		
	RV4	0.927		
Process Performance (PC)	PC1	0.827	0.749	0.8
	PC2	0.902	1	56
Product Performance (PD)	PD1	0.770	0.749	0.898
	PD2	0.828		
	PD3	0.916		
	PD4	0.795		

Table 4 Discriminant validity of the constructs.

Constructs	EI	II	PC	PD	RV	ST
External Integration	0.841					
Internal Integration	0.455	0.780				
Process Performance	0.141	0.453	0.866			
Product Performance	-0.175	0.133	0.129	0.829		
Requirement Volatility	0.123	-0.051	-0.490	-0.278	0.724	
Standardization	0.482	0.438	0.157	0.269	-0.085	0.730

5.2 Structural Model Evaluation

The next step after the measurement model evaluation was assessing the structural model, which includes testing of the theoretical hypotheses and assessment of the relationships between the latent constructs. SEM allows for simultaneous testing of the hypotheses [44]. Assessment of the structural model was conducted by estimating the path coefficient and t-value for each hypothesized path. The path coefficients are equivalent to the standardized partial regression coefficients of multiple regression and represent the effect of one variable on another dependent variable. Path analysis can be used to test how well a hypothetical model fits the data. The overall results are presented in Figure 1.

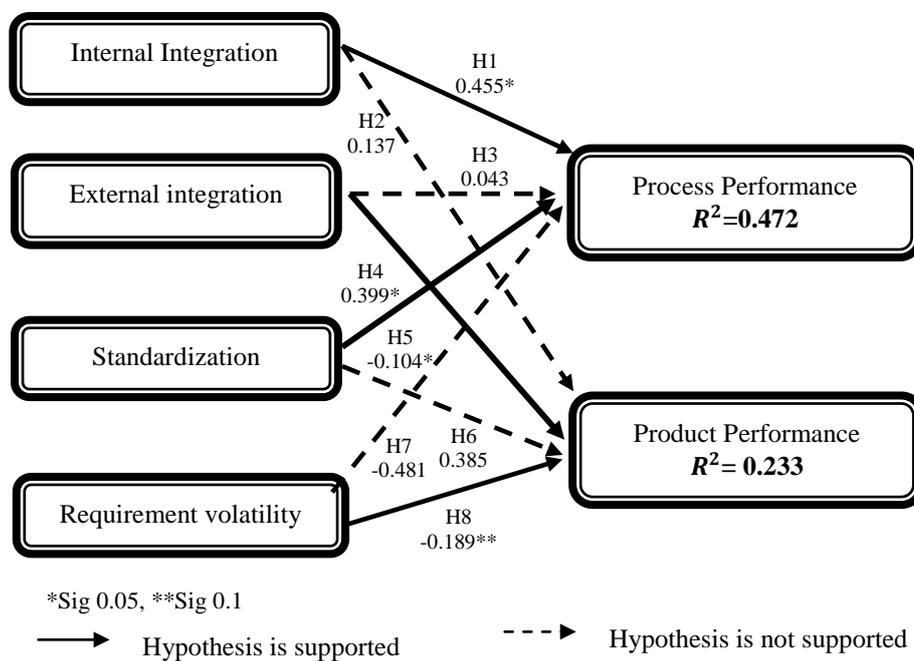


Figure 1 Results of hypotheses testing.

Four out of eight hypotheses developed in this study were supported. Internal integration and standardization significantly influenced the process performance of the ISD projects and therefore H1 and H5 were supported. External integration and requirement volatility also had a significant influence on the product performance of the ISD projects and therefore H4 and H8 were supported. External integration and requirement volatility did not significantly influence process performance and hence H3 and H7 were not supported. The influence of internal integration and standardization on product performance were found not significant and thus H2 and H6 were not supported.

6 Discussion

This study found that internal integration has a positive influence on process performance in ISD projects and has no effect on the performance of the product delivered in a project. After further examination regarding the effect of internal integration on product performance, it was found that the path coefficient from internal integration towards product performance was greater when the path went through external integration. This means that internal integration cannot directly affect the performance of the product, but the effect is mediated by external integration. A high degree of external integration indicates a high level of client participation and this high involvement, if not managed properly, may lead to a high level of conflict between the key involved people on the client side and a conflict between the client's people and analysts on the vendor side. This high level of conflict may potentially reduce the quality of the working atmosphere and the morale of the project team members, which in turn may decrease the quality of the product delivered in the project. Managers from the client company should be very careful in selecting the right key users to be involved in a project.

As a knowledge intensive process, ISD needs the integration of technological and business process knowledge. The importance of knowledge integration is in line with the research conducted by Park and Lee [47], who found that knowledge sharing can contribute to the success of IT/IS outsourcing activities. Further, Park and Lee [47] stated that knowledge sharing activities can encourage project members to become more innovative and creative. In the context of offshore software development (OSD) projects, Xu and Yao [48] found that the vendors' relationship with the client helps to overcome challenges in the software development process, mediated by knowledge sharing among the actors involved in the process. To foster the relationship between client and vendor, communication capability is believed to be a very important factor [49].

Standardization of control has a positive influence on the performance of the process, but has no effect on the performance of the product. While previous studies on outsourced ISD projects have not examined the direct influence of standardization on product performance, this study found that the standardization has no significant influence on product performance. Requirement volatility has a negative effect on product performance but does not negatively affect process performance.

7 Conclusion

Analyzing the overall results of the hypotheses testing, process performance is significantly influenced by internal integration and standardization, while product performance is significantly influenced by external integration and requirement volatility. These results show that the role of the project team members and other involved stakeholders from the client company is especially important for ensuring product performance in ISD projects. On the other hand, the way in which team members work together, especially on the vendor's side, plays an important role in ensuring process performance.

8 Limitations and Future Research

In this study, project performance and product performance were measured using the vendors' perceptions. Future studies should also take the perception of the client companies into consideration, as the party that receives the product development results. Involving the vendor as well as the client in the data collection process is expected to produce more objective data. The second limitation of this study was that it did not provide any analysis associated with characteristics of ISD projects such as the type of systems or applications that are developed, the duration of the project development, project team size, the type of outsourcing scheme, and the type of client companies. Future studies can examine the effects of differences in project characteristics on project success.

References

- [1] Ropponen, J., *Software Risk Management – Foundations, Principles and Empirical Findings*, Jyvaskyla University Printing House, 1999.
- [2] Nidumolu, S.R., *The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable*, *Information Systems Research*, **6**(3), pp. 191-219, 1995.
- [3] Nidumolu, S.R., *Standardization, Requirement Uncertainty and Software Project Performance*, *Information and Management*, **31**(3), pp. 135-150, 1996.
- [4] Zowghi, D. & Nurmaliani, N., *A Study of the Impact of Requirement volatility on Software Project Performance*, in Ninth Asia-Pacific Software Engineering Conference (APSEC'02), 2002.
- [5] Na, K.S., *The Impacts of Requirement Uncertainty and Standardization on Software Project Performance: A Comparison of Korea and USA*, *Journal of Information Technology Applications & Management*, **11**(2), pp. 15-27, 2004.

- [6] Na, K.S., Li, X., Simpson, J.T. & Kim, K.Y., *Uncertainty Profile and Software Project Performance: A Cross-National Comparison*, Journal of Systems and Software, **70**(1), pp. 155-163, 2004.
- [7] Na, K.S., Simpson, J.T., Li, X., Singh, T. & Kim, K.Y., *Software Development Risk and Project Performance Measurement: Evidence in Korea*, Journal of Systems and Software, **80**(4), pp. 596-605, 2007.
- [8] Glen, P., *Detecting Disaster Projects*, Computer World, **40**(6), 2006.
- [9] Barki, H., Rivard, S. & Talbot, J., *An Integrative Contingency Model of Software Project Risk Management*, Journal of Management Information Systems, **17**(4), pp. 37-69, 2001.
- [10] Wallace, L., Keil, M. & Rai, A., *Understanding Software Project Risk: A Cluster Analysis*, Information and Management, **42**(1), pp. 115-125, 2004.
- [11] Wang, Q.Z. & Liu, J., *Project Uncertainty, Management Practice and Project Performance: An Empirical Analysis on Customized Information Systems Development Projects*, Engineering Management Conference, IEEE International, pp. 341-345, 2006.
- [12] Jiang, J.J., Klein, G. & Chen, H.G., *The Effects of User Partnering and User Non-Support on Project Performance*, Journal of the Association for Information Systems, **7**(2), pp. 68-90, 2006.
- [13] Jun, L., Qinzen, W. & Qingguo, M., *The Effects of Project Uncertainty and Risk Management on IS Development Project Performance: A Vendor Perspective*, International Journal of Project Management, **29**(7), pp. 923-933, 2010.
- [14] Patnayakuni, R., Ruppel, C.P. & Rai, A., *Managing the Complementarity of Knowledge Integration and Process Formalization for Systems Development Performance*, Journal of the Association for Information Systems, **7**(8), pp. 545-567, 2006.
- [15] Purvis, R.L., Sambamurthy, V. & Zmud, R.W., *The Assimilation of Knowledge Platforms in Organizations: An Empirical Investigation*, Organization Science, **12**(2), pp. 117-135, 2001.
- [16] Patnayakuni, R., Rai, A. & Tiwana, A., *Systems Development Process Improvement: A Knowledge Integration Perspective*, IEEE Transactions on Engineering Management, **54**(2), pp. 286-300, 2007.
- [17] Parolia, N., Goodman, S., Li, Y. & Jiang, J.J., *Mediators Between Coordination and IS Project Performance*, Information and Management, **44**(7), pp. 635-645, 2007.
- [18] Lacity, M. & Willcocks, L.P., *An Empirical Investigation of Information Technology Sourcing Practice: Lessons from Experience*, MIS Quarterly, **22**(3), pp. 363-408, 1998.
- [19] Dey, P.K., Kinch, J. & Ogunlana, S.O., *Managing Risk in Software Development Projects: A Case Study*, Industrial Management and Data Systems, **107**(2), pp 284-303, 2007.

- [20] Stark, G., Skillicorn, A. & Ameen, R., *An Examination of the Effects of Requirements Changes on Software Releases*, CROSSTALK The Journal of Defence Software Engineering, December 1998.
- [21] Ferreira, S., Collofello, J., Shunk, D. & Mackulak, G., *Understanding the Effects of Requirement Volatility in Software Engineering by Using Analytical Modeling and Software Process Simulation*, The Journal of Systems and Software, **82**(10), pp. 1568-1577, 2009.
- [22] Kidd, P.T., *Design of Human-Centred Robotic Systems*, in *Human-Robot Interaction*, edited by M. Rahimi and W. Karwowski, 1992.
- [23] Humphrey, W.S., *Managing the Software Process*, Addison-Wesley, Reading, MA, 1989.
- [24] Carlile, P., *Transferring, Translating, and Transforming: An Integrative Framework for Managing Knowledge across Boundaries*, Organization Science, **15**(5), pp. 555-568, 2004.
- [25] Tiwana, A. & Keil, M., *Does Peripheral Knowledge Complement Control? An Empirical Test in Technology Outsourcing Alliances*, Strategic Management Journal, **28**(6), pp. 623-634, 2007.
- [26] Postrel, S., *Islands of Shared Knowledge: Specialization and Mutual Understanding in Problem Solving Teams*, Organization Science, **13**(3), pp. 303-320, 2002.
- [27] Eisenhardt, K.M. & Tabrizi, B.N., *Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry*, Administrative Science Quarterly, **40**(1), pp. 84-110, 1995.
- [28] Zirger, B.J. & Hartley, J.L., *The Effect of Acceleration Techniques on Product Development Time*, IEEE Transactions on Engineering Management, **41**(5), pp. 143-152, 1996.
- [29] Griffin, A., *The Effect of Project and Process Characteristics on Product Development Cycle Time*, Journal of Marketing Research, **34**(1), pp. 24-35, 1997.
- [30] Yetton, P., Martin, A., Sharma, R. & Johnston, K., *A Model of Information Systems Development Project Performance*, Information Systems, **10**(4), pp. 263-289, 2000.
- [31] Ewusi-Mensah, K., *Critical Issues in Abandoned Information Systems Development Projects*, Communications of the ACM, **40**(9), pp. 74-80, 1997.
- [32] Brown, S. & Eisenhardt, K., *Product Development: Past Research, Present Findings, and Future Directions*, Academy of Management Review, **20**(2), pp. 343-378, 1995.
- [33] Ives, B. & Olson, M., *User Involvement and MIS Success: A Review of Research*, Management Science, **30**(5), pp. 586-603, 1984.
- [34] Paulk, M.C., Curtis, B., Chrissis, M.B. & Weber, C.V., *Capability Maturity Model (Version 1.1)*, IEEE Software, **10**(4), pp. 18-27, 1993.

- [35] Henderson, J.C. & Lee, S., *Managing I/S Design Teams: A Control Theories Perspective*, Management Science, **38**(6), pp. 757-777, 1992.
- [36] Suh, C.K. & Jeong E.H., *The Effect of Project Risk and Risk Management on Software Development Project Performance*, Asia Pacific Journal of Information Systems, **13**(2), pp. 199-217, 2003.
- [37] Liu, J.Y.C., Chen, J.V., Chan, C.L. & Lie, T., *The Impact of Software Process Standardization on Software Flexibility and Project Management Performance: Control Theory Perspective*, Information and Software Technology, **50**(9), pp. 889-896, 2008.
- [38] Tiwana, A., Wang, J., Keil, M. & Ahluwalia, P., *The Bounded Rationality Bias in Managerial Valuation of Real Options: Theory and Evidence from IT Projects*, Decision Sciences, **38**(1), pp. 157-181, 2007.
- [39] Lane, M. & Cavaye, A.L.M., *Management of Requirement volatility Enhances Software Development Productivity*, in Third Australian Conference on Requirements Engineering (ACRE 98), Geelong, Australia, 1998.
- [40] Zowghi, D. & Nurmuliani, N., *Investigating Requirement volatility During Software Development: Research in Progress*, in Third Australian Conference on Requirements Engineering (ACRE 98), Geelong, Australia, 1998.
- [41] Zowghi D., Offen, R. & Nurmuliani, N., *The Impact of Requirement volatility on Software Development Lifecycle*, in International Conference on Software, Theory and Practice (ICS2000), Beijing, China, 2000.
- [42] Rai, A. & Hindi, A.H., *The Effects of Development Process Modeling and Task Uncertainty on Development Quality Performance*, Information & Management, **37**(6), pp. 335-346, 2000.
- [43] Ullman, J.B., *Structural Equation Modeling*, In B.G. Tabachnick & L.S. Fidell (Eds.), Using Multivariate Statistics (4th ed.), Needham Heights, MA: Allyn & Bacon, 2001.
- [44] Hair, J.F., Black, W.C., Babin, B.J. & Anderson, R.E., *Multivariate Data Analysis*, Seventh Edition, Prentice Hall, Upper Saddle River, New Jersey, 2010.
- [45] Fornell, C., & Larcker, D.F., *Evaluating Structural Equation Models with Unobservable Variables and Measurement Error*, Journal of Marketing Research, **18**(1), pp. 39-50, 1981.
- [46] Gefen, D., Straub, D.W. & Boudreau, M.C., *Structural Equation Modeling Techniques and Regression: Guidelines for Research Practice*, Communications of the AIS, **1**(7), pp. 1-78, 2000.
- [47] Park, J.G. & Lee, J., *Knowledge Sharing in Information Systems Development Projects: Explicating the Role of Dependence and Trust*, International Journal of Project Management, **32**, pp. 153-165, 2014.

- [48] Xu, P. & Yao, Y., *Knowledge Sharing in Offshore Software Development: A Vendor Perspective*, Journal of Global Information Technology Management, **16**(1), pp. 58-84, 2013.
- [49] Swar, B., Moon, J., Oh, J., & Rhee, C., *Determinants of Relationship Quality for IS/IT Outsourcing Success in Public Sector*, Information Systems Frontiers, **14**(2), pp. 457-475, 2012.