

Risk Factors for Software Requirements Change Implementation

Marfizah A.Rahman¹, Rozilawati Razali², Fatin Filzahti Ismail³

Center for Software Technology and Management, Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia, Malaysia

Abstract—Requirements change has been regarded as a substantial risk in software development projects. The factors that contribute to the risk are identified through impact analysis, which later determine the planning of the change implementation. The analysis is however not straightforward as the risk factors that constitute requirements change implementation is currently not much explored. This paper identifies the risk factors by firstly collating them qualitatively through a review of related work and a focus group study. The factors are then confirmed quantitatively through a survey in which data is analysed by using Partial Least Squares Structural Equation Modelling (PLS-SEM). The survey comprise of 276 practitioners from software industry who are involved in the impact analysis. The results indicate that User, Project Team, Top Management, Third Party, Organisation, Identification of Change, Existing Product and Planning of Change Implementation are the significant risk factors in planning of requirements change implementation.

Keywords—Requirements change; risk factor; structural equation modeling

I. INTRODUCTION

Changes in system requirements have negative and positive impacts on system development projects. The changes will have a negative impact if the processes are not properly managed [1]. Errors in managing the change requirements may result in project's cost increases, delay in the system development and affecting the system quality [2]. On the other hand, the process will have a positive impact if the changes conducted to the system are successfully implemented. System improvements can improve the level of user trust in the system [3].

Requirements change is one of the crucial risks in software projects [4], [5]. Requirements changes are inevitable in software system development projects. The addition of new requirements and changes to the existing requirements can occur at all stages of the system development process [6]. New requirements come in tandem with technological developments. As such, consumer needs are changing to meet the organisation's policies and operating and business operations environments. The changes occur due to several factors such as errors in original requirements, evolving customer needs, technological changes, and changes in the business environment or organisational policy [6].

Requirements change is proposed through change requests. Dealing with requirements change requests is a decision-intensive process [7], as it involves cost and resources [8].

Deciding whether or not to accept the change request is done by an important team called Change Control Board (CCB). CCB conducts impact analysis before the decision is made by comparing cost and risk [9], [10]. Impact analysis has great benefit in reducing the risks of implementing requirements change [11], [12]. By identifying potential impacts before making a change, the risks of embarking on a costly change can be greatly reduced [11], [13]. To ensure an accurate decision, CCB should be able to analyse the impacts of the identified factors holistically and assess their risk levels. Also, in order to ensure informed decision during impact analysis, CCB needs a predetermined risk factors or criteria to evaluate a requirements change request [14].

The risk factors to be considered in assessing the requirements change requests are however still indeterminate. Previous studies have shown that there are several risk factors that influence the change requirements and thus contribute to the successfulness of their implementation [15], [16]. The studies are more focused on technical aspects [17], instead of the non-technical aspects [18]. This resulted in unreliable and inaccurate decision making in the change request implementation.

Hence, the aim of this paper is to present the risk factors of the requirements change implementation. It focuses on change impact elements. The study uses a partial least square structural equation modelling (SEM) technique. The identified risk factors discussed in this paper were acquired through series of work conducted previously, reported in [19].

The paper is organized as follows: Section 2 provides the related work on the subject matter. Section 3 elaborates the research model and hypothesis. Section 4 briefly explains the methodology used. Section 5 provides the results. Finally, Section 6 concludes the paper with a summary that outlines the main findings and future work.

II. RELATED WORK

Requirements change has been recognised as a major issue in software development projects. One main issue is about how to assess a requirements change request before deciding either to accept or reject it for implementation [20]. The decision has to be made after impact analysis [21], [22], by which the affected elements are identified and scrutinised [23].

Over the past decade, impact analysis has become a major concern among practitioners and researchers in software engineering [8], [20], [24], [25]. For instance, a study has

proposed a tool to ensure an accurate impact analysis by identifying the hardware and software in the current system that are likely to be impacted by the proposed change [26], [54]. Another study developed a tool to determine the planning strategies in assigning human resources to change activities [27]. Both tools however have lack of functions which are to store and retrieve information of past changes [28].

The system is developed for users and based on user requirements. Change requests are triggered by users based on the experiences that they had with the system. When requesting a change, the users must provide information about the current system and clarify the change that they request to the project team [29]. Users must be involved in the process so that they are aware of the system operations after the requested change has been made [30], [54]. Among the identified risks involved is the less user involvement during systems development and inadequate user's knowledge regarding the developed systems [31]. The users also often lack of cooperation with the project team when needed [31], [32], less commitment towards the systems and has a negative attitude towards the system [32].

Later, project team is responsible to bring forward the request to CCB meeting for them to analyse and assess its impacts [54]. The project team plays an important role in system development. Among the risks involved with the project team are low-skilled capability in the development of the system [32]. As such, new members that are involved in the project may have less knowledge about the system that is being developed [32]. This might be due to the limited training given to the project team [33]. Besides that, the project team is also identified to have less experience in system development projects [31]. Experience gives the project team confidence to develop systems. On the other hand, there is also issue regarding the project team's lack of commitment in the system development [31]. During impact analysis, it is important to assess the project team's capability such as experience, motivation, skill and knowledge in order to ensure the feasibility of the change [34].

Besides that, support and commitment from the top management are also essential [31], [32]. Top management is responsible for the implementation of system development. The top management commitment is needed to make the decision of a subject matter. The absence of specific communication channel among the top management may lead to lack of commitment in the project [32]. Besides that, support from top management is also crucial. It is identified that there are top managements that failed to give support when needed, affecting the change requirement process [31].

Furthermore, the degree of dependency on the third party [32] and their involvement in the project [31] also contribute to the risks. The project development also involves third parties such as suppliers and external organisations related to the system. The risk factors that arise are the number of third parties involved [31]. When the number of third parties is large, related management becomes difficult. In addition, the high levels of dependence by the project team to third parties also pose a risk to system development projects [32].

The change request might affect the existing software components such as source code [35], [36], documentation [37], tools [38] and architecture [36], as well as the hardware components such as memory usage [39], performance [12] and platform [40]. There are also risks of complex architecture [4]. Out of all the components, the technology component is identified to have highest risk. Due to that reason, the analysis of a change request must consider not only the people involved but also the hardware and software used in the current system.

The costs and schedules are also important for project implementation. Determining the estimated cost and the schedule is not easy [32]. The cost to implement requirements change is influenced by the number of project team members and consultants as well as project duration, size and scope [41]. In fact, the incorrect or insufficient estimation of cost and schedule may contribute to more risk to the project [31]. On the other hand, optimising the schedule for implementing any requirements change schedule can result in significant time saving [42]. Therefore, it is important to judge the affected and affecting elements accurately to ensure an adequate allocation of time and cost in implementing the change [33], [31]. Strategic planning and technology standards imposed by the organisation also have been found to influence the change implementation [19], [43].

The goal of requirements change is to improve the value of the current software system. However, it triggers the risks in terms of late delivery, cost overrun, low product quality and sometimes failure to the entire software project [44]. As requirements change effort is considered as a project, the risk factors concerning projects also apply to them. Some instances of project risks are inexperienced project team members [33], commitment from team members towards the project and effective communication between team members and users [31].

The review above indicates that there are various elements that are deemed necessary when managing requirements change. As the elements contribute to the success or failure of a requirements change project, they are considered as risk factors. To date, it is uncertain how CCB should analyse the impacts of those risks and subsequently decide the way forward [45]. There are interrelationships between the risks [46]. The significance of these interrelationships needs to be confirmed objectively [4], [5] so that an accurate impact analysis can be made [20].

III. METHODOLOGY

The methodology for this study is the combination of qualitative and quantitative approach (mixed-method) [47], [48]. Fig. 1 illustrates the research design which contains the main activities involved in the study. The qualitative part consists of literature review and focus group study which have been conducted earlier to gather the identified risk factors as well as to confirm them. The findings have been reported in [19]. However, the findings from empirical study was only validated solely through opinion-based perspective, thus it should be refined and strengthen further by confirming the risk factors quantitatively through a large scale survey.

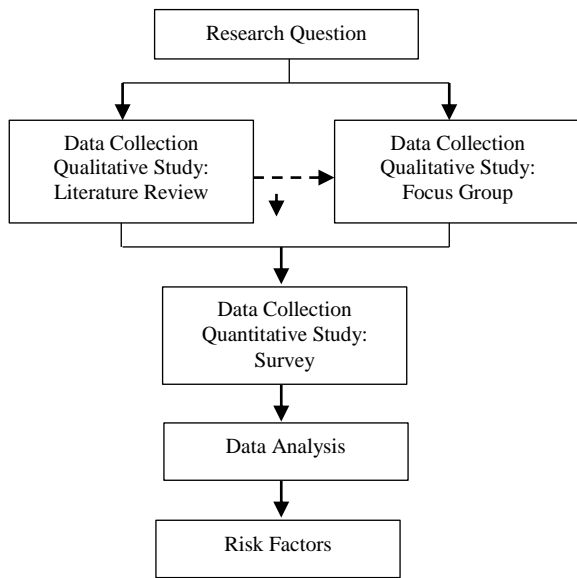


Fig. 1. Research Design.

Hence, a survey was conducted to confirm the existence and significance of the risk factors as the continuation work of one of the stage in the study. The questionnaires consisted of 37 questions covering the identified risk factors. The questions used a 7-point Likert agreement scale from (1) strongly not important to (7) strongly important. A panel of software experts validated the questions. Then, a pilot study was conducted involving 50 software practitioners from the University’s Computer Centre and 10 postgraduate students from the University. The questionnaires were then improved based on the findings received from both exercises.

TABLE I. DEMOGRAPHIC INFORMATION OF RESPONDENTS

Demographic Information	Percentage (%)
Organisation Type	
Government agencies	69.9
Semi-government	17.8
Private companies	12.3
Role	
Project managers	12.0
Project leaders	13.4
Project teams	72.1
Others	2.5
Working Experience Duration	
Less than 5 years	43.8
5-10 years	37.7
11–15 years	10.5
More than 15 years	8.0
Number of project	
Less than 5 projects	51.4
5-10 projects	29.0
More than 10 projects	19.6

The questionnaires were distributed to a number of information technology (IT) and software organisations in Malaysia within the period of 3 months. The organisations comprised of public and private sectors as well as local and multinational companies. There were 400 questionnaires disseminated but only 287 questionnaires were returned. However, 11 out of 287 questionnaires-were omitted from the analysis due to incomplete answers. Therefore, analysis was made towards a number of 276 questionnaires which were answered completely. This caused the response rate to be 69.

The demographic information of the 276 respondents are listed in Table I.

The collected data were then analysed by using Partial Least Squares Structural Equation Modelling (PLS-SEM).

IV. MODEL AND HYPOTHESES

Fig. 2 below illustrates a conceptual model of requirements change implementation which was based on previous conceptual model work [19]. The conceptual model was updated with the empirical output from the focus group to produce the model. The model contains eight essential components: User, Project Team, Top Management, Third Party, Organisation (Strategic Planning, Technology Standard), Identification of Change, Existing Product (Software, Hardware) and Planning of Change Implementation.

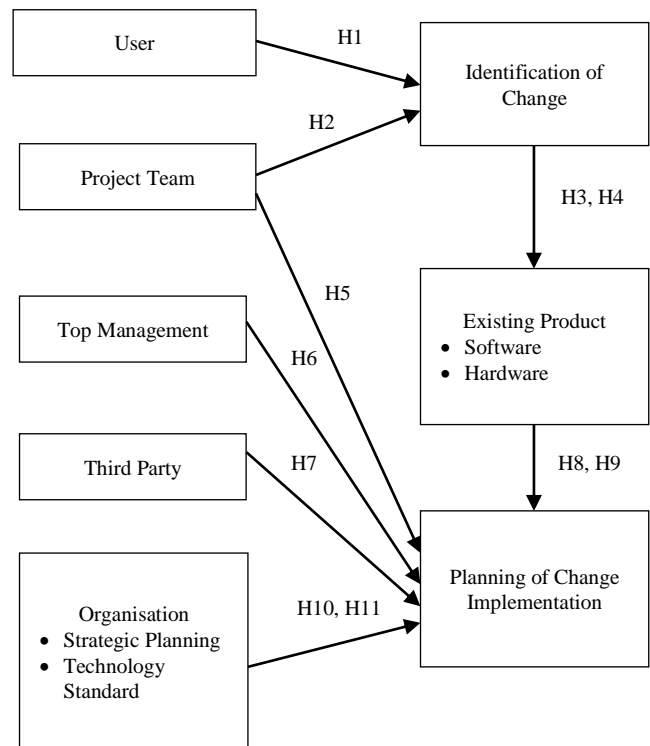


Fig. 2. Conceptual Model of Requirements Change Implementation.

Based on the conceptual model, the hypotheses were derived as depicted in Table II. The related works that support the hypotheses are also included in the table.

TABLE II. HYPOTHESES

	Hypothesis	Related Work
H1	User is positively associated with Identification of Change	[33][31][54]
H2	Project Team is positively associated with Identification of Change	[33][31][54]
H3	Project Team is positively associated with User	[33]
H4	Identification of Change is positively associated with Software	[32]
H5	Identification of Change is positively associated with Hardware	[32], [54]
H6	Software is positively associated with Planning of Change Implementation	[32]
H7	Hardware is positively associated with Planning of Change Implementation	[32]
H8	Software is positively associated with Hardware	[32]
H9	Project Team is positively associated with Planning of Change Implementation	[33][31]
H10	Top Management is positively associated with Planning of Change Implementation	[31]
H11	Third Party is positively associated with Planning of Change Implementation	[31]
H12	Strategic Planning is positively associated with Planning of Change Implementation	[19]
H13	Technology Standard is positively associated with Planning of Change Implementation	[19]

V. RESULTS

The common analysis conducted in PLS-SEM approach comprises of two stages [47]. The first stage is the analysis of measurement model. The purpose of measurement model is to examine the relationships between the latent constructs and their measures. This is to ensure the reliability and validity of the model. On the other hand, the goal of structural model is to assess the relationships between the latent constructs. The following paragraphs explain each model and its analysis, respectively.

A. Measurement Model

The measurement model refers to the relationships between the constructs and the items used to measure them. The analysis evaluates the convergent validity and discriminant validity of the constructs [48]. The convergent validity of a construct is established when the reliability level of the individual items that measure the construct is above 0.70, the composite reliability (CR) of the construct exceeds 0.70 and the average variance extracted (AVE) of the construct exceeds 0.5 [49]. The discriminant validity of a model is assessed by comparing each construct's square root of AVE against its bivariate correlations with other constructs as well as checking cross-loading cases [50].

The results of measurement model are shown in Table III. The descriptions of items used for the measurement are listed in Table IV. According to [51], the factor loading for items

that exceed the level of 0.6 is acceptable in exploratory research, whereas the factor loading of 0.70 to 0.90 is satisfactory [52] and above 0.90 indicates that the items measure the phenomenon accurately. The table shows that most items are satisfactory and above. There are only two items below 0.70, namely PT4 and PL2. They are still retained because they are within the acceptable level.

Convergent validity is the degree to which multiple items measure the same construct under the study. The common approach to examine convergent validity is the factor loadings and AVE. In this study, the AVE values are greater than 0.50. This suggests convergent validity is at the construct level. In addition, the CR values are greater than 0.70. This indicates the reliability is acceptable.

Discriminant validity is the degree to which the measures of different concepts are distinct. The discriminant validity is established between two constructs if the square root of AVE of each one is higher than the shared variance. The shared variances are compared to AVEs as shown in Table V. Since the AVE values of the two constructs are higher than the squared correlation, the discriminant validity among the latent constructs is supported [51].

TABLE III. MEASUREMENT MODEL

Construct	Item	Convergent Validity		
		Loading	CR	AVE
Top Management	TM1	0.942	0.884	0.792
	TM2	0.834		
Third Party	TP1	0.915	0.927	0.864
	TP2	0.944		
Technology Standard	TG1	0.973	0.972	0.945
	TG2	0.972		
Project Team	PT1	0.840	0.899	0.641
	PT2	0.842		
	PT3	0.782		
	PT4	0.670		
	PT5	0.853		
Planning	PL1	0.718	0.856	0.602
	PL2	0.631		
	PL3	0.871		
	PL4	0.858		
Identification	ID1	0.855	0.918	0.789
	ID2	0.910		
	ID3	0.898		
User	US1	0.830	0.894	0.629
	US2	0.820		
	US3	0.786		
	US4	0.819		
	US5	0.703		
Software	SW1	0.801	0.898	0.596
	SW2	0.784		
	SW3	0.707		
	SW4	0.780		
	SW5	0.717		
	SW6	0.835		
Hardware	HW1	0.902	0.895	0.826
	HW2	0.911		
	HW3	0.914		
Strategic Planning	SP1	0.941	0.842	0.863
	SP2	0.917		

TABLE IV. ITEM DESCRIPTION

Construct	Item	Item Description
Top Management	TM1 TM2	Commitment Support
Third Party	TP1 TP2	Number of third party involved Dependency to external agents
Technology Standard	TG1 TG2	Software Hardware
Project Team	PT1 PT2 PT3 PT4 PT5	Skill Commitment Motivation Experience Knowledge
Planning	PL1 PL2 PL3 PL4	Schedule Cost Hardware Resource
Identification	ID1 ID2 ID3	Reasons Type Resource
User	US1 US2 US3 US4 US5	Cooperation Readiness Commitment Knowledge Involvement
Software	SW1 SW2 SW3 SW4 SW5 SW6	Tools Interface Documentation Integration Source code Architecture
Hardware	HW1 HW2 HW3	Memory Platform Performance
Strategic Planning	SP1 SP2	Goal Policy

B. Structural Model

The structural model describes the dependency relationships between the constructs, as presented in Fig. 2. The analysis was done on the explanatory capacity of the model and the statistical significance of the various structural factors. Based on the results, the coefficient of determination (R^2) value of planning (0.4879) was considered as moderate [53]. However, R^2 value of change identification (0.2206), software (0.2725) and hardware (0.2020) was weak.

Apart from R^2 , Q^2 coefficient was also used as a threshold for predictive relevance [51]. The Q^2 coefficient in PLS-SEM was generated from a technique called blindfolding. The generated result of Q^2 for this model was 0.2927. Q^2 values were considerably above zero. This indicates the model's predictive relevance for the construct [51].

After evaluating the explanatory capacity of the structural model, the statistical significance of the various structural coefficients was tested through technique called bootstrapping to generate t-statistic value associated with each path. The outcomes are showed in Table VI.

The results showed that User and Project Team had significant relationship with Identification of Change. These results were similar with [33]. It means that the identification of change highly depended on user and project team. The Identification of Change also had significant relationships with Hardware and Software, which were also supported by [39].

The Project Team, Top Management, Strategic Management, Hardware and Software had significant relationships with Planning of Change Implementation. It means that planning of change implementation should consider the impact of change on project team, top management, strategic management, hardware and software. On the other hand, the relationship between Third Party and Technology Standard with Planning of Change Implementation was not significant.

TABLE V. DISCRIMINANT VALIDITY

	TM	TG	TP	PT	ID	US	PL	SW	HW	SP
Top Management (TM)	0.890									
Technology Standard (TG)	0.439	0.972								
Third Party (TP)	0.155	0.183	0.930							
Project Team (PT)	0.426	0.429	0.080	0.800						
Identification (ID)	0.382	0.436	0.120	0.442	0.888					
User (US)	0.387	0.313	0.070	0.589	0.348	0.793				
Planning (PL)	0.475	0.477	0.152	0.491	0.564	0.404	0.776			
Software (SW)	0.384	0.381	0.110	0.430	0.517	0.372	0.526	0.772		
Hardware (HW)	0.310	0.422	0.115	0.297	0.417	0.293	0.517	0.577	0.909	
Strategic Planning (SP)	0.347	0.541	0.178	0.436	0.429	0.244	0.470	0.339	0.298	0.929

TABLE VI. RESULTS OF PATH COEFFICIENT AND T-VALUE

Hypothesis	Path Relationship	Path Coefficient	t-value	Decision
H1	US → ID	0.135	2.005	Supported
H2	PT → ID	0.362	5.879	Supported
H3	PT → US	0.589	13.360	Supported
H4	ID → SW	0.517	9.242	Supported
H5	ID → HW	0.162	2.408	Supported
H6	SW → PL	0.163	2.649	Supported
H7	HW → PL	0.236	3.370	Supported
H8	SW → HW	0.493	7.585	Supported
H9	PT → PL	0.172	2.916	Supported
H10	TM → PL	0.173	2.733	Supported
H11	TP → PL	0.025	0.637	Not Supported
H12	SP → PL	0.167	2.348	Supported
H13	TG → PL	0.070	0.885	Not Supported

The factors that are significant are important during impact analysis. Therefore, they are considered as risk factors for requirements change implementation. This finding provides empirical evidence that the risks factors can decrease the degree of project success [31].

VI. CONCLUSION AND FUTURE WORK

This study aimed to confirm the risk factors for requirements change implementation. A survey was conducted to collect the data, which were then analysed using PLS-SEM approach. The results showed that user, project team, top management, identification of change, strategic management, hardware and software were significant factors for planning of change implementation. On the other hand, two factors were found not significant for the planning of change implementation; namely the third party and technology standard. As these findings were obtained through quantitative study, the reason of the insignificant results could not be determined clearly. Hence, future work on qualitative approach could be conducted to investigate the underlying reason behind those opinions.

The analysis focused on impact analysis. Therefore, the analysis can be replicated to other determinant risk factors for requirements change implementation. The model could then be extended as a risk measurement model for requirements change initiatives.

ACKNOWLEDGMENT

This work was funded by Fundamental Research Grant Scheme (FRGS/2/2013/ICT01/UKM/02/02).

REFERENCES

- [1] V. Yadav and M. Adya, "Control , Process Facilitation , and Requirements Change in Offshore Requirements Analysis : The Provider Perspective," *Jounal Inf. Technol. Theory Appl.*, vol. 14, no. 3, pp. 30–47, 2013.
- [2] Z. Jiayi, L. Yunjuan, and G. Yuesheng, "The Requirements change Analysis for Different level users," in *International Symposium on Intelligent Information Technology Application Workshops*, 2008, pp. 987–989.
- [3] P. Palvia, A. Patula, and J. Nosek, "Problems and Issues in Application Software Maintenance Management," *J. Inf. Technol. Manag.*, vol. 4, pp. 17–28, 1995.
- [4] Y. Fu, "Impact Propagation and Risk Assessment of Requirement Changes for Software Development Projects based on Design Structure Matrix," *Int. J. Proj. Manag.*, vol. 30, no. 3, pp. 363 – 373., 2012.
- [5] M. R. Strens and Sugden, "Change Analysis : A Step towards Meeting the Challenge of Changing Requirements," in *IEEE*, 1996, pp. 278–283.
- [6] N. Nurmuliiani, D. Zowghi, and S. P. Williams, "Requirements Volatility and Its Impact on Change Effort : Evidence-based Research in Software Development Projects," *AWRE 2006*, 2006.
- [7] A. Aurum and C. Wohlin, "The fundamental nature of requirements engineering activities as a decision-making process," *Inf. Softw. Technol.*, vol. 45, no. 14, pp. 945–954, Nov. 2003.
- [8] B. B. Chua and J. Verner, "Examining Requirements Change Rework Effort: A Study," *Int. J. Softw. Eng.*, vol. 1, no. 3, pp. 48–64, Jul. 2010.
- [9] M. Sirola, "Conceptual Decision Model," *IEEE Int. Work. Intell. Data Acquis. Adv. Comput. Syst. Technol. Appl.*, pp. 69–72, 2003.
- [10] J. Li, M. Li, D. Wu, and H. Song, "An integrated risk measurement and optimization model for trustworthy software process management," *Inf. Sci. (Ny).*, vol. 191, no. 15, pp. 47–60, May 2012.
- [11] S. I. Mohamed, I. a. M. Elmaddah, and A. M. Wahba, "Hybrid-Based Maintainability Impact Analysis for Evolving Systems," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 19, no. 08, pp. 1131–1149, Dec. 2009.
- [12] B. J. Williams and P. B. J. Williams, "Change Risk Assessment : Understanding Risks Involved in Changing Software Requirements," *2006 Int. Conf. Softw. Eng. Res. Pract.*, 2006.
- [13] R. S. Arnold and S. a. Bohner, "Impact analysis-Towards a framework for comparison," *IEEE*, pp. 292–301, 1993.
- [14] P. Rovegard, L. Angelis, and C. Wohlin, "An Empirical Study on Views of Importance of Change Impact Analysis Issues," *IEEE Trans. Softw. Eng.*, vol. 34, no. 4, pp. 516–530, 2008.
- [15] Y. Li, J. Li, Y. Yang, and M. Li, "Requirement-Centric Traceability for Change Impact Analysis: A Case Study," *Springer-Verlag Berlin Heidelb.*, pp. 100–111, 2008.
- [16] S. Park, H. Kim, and D. H. Bae, "Impact Analysis of a Software Process Using Process Slicing," *2009 Ninth Int. Conf. Qual. Softw.*, 2009.
- [17] B. Michalik and D. Weyns, "Towards a Solution for Change Impact Analysis of Software Product Line Products," *Ninth Work. IEEE/IFIP Conf. Softw. Archit.*, pp. 290–293, 2011.
- [18] B. J. Williams and J. C. Carver, "Characterizing Software Architecture Changes : A Systematic Review," *Inf. Softw. Technol.*, vol. 52, pp. 31–51, 2010.
- [19] M. A. Rahman, R. Razali, and D. Singh, "A Risk Model of Requirements Change Impact Analysis," *J. Softw.*, vol. 9, no. 1, pp. 76–81, 2014.
- [20] H. O. Ali, M. Z. A. Rozan, and A. M. Sharif, "Identifying challenges of change impact analysis for software projects," *Int. Conf. Innov. Manag. Technol. Res.*, pp. 407–411, May 2012.
- [21] M. W. Bhatti, F. Hayat, and S. Ahmed, "A Methodology to Manage the Changing Requirements of a Software Project.," in *Information Systems*, 2010, pp. 319–322.
- [22] B. B. Chua and J. Verner, "Examining Requirements Change Rework Effort: A Study," *Int. J. Softw. Eng.*, vol. 1, no. 3, pp. 48–64, 2010.

- [23] S. Park and D. Hwan Bae, "An Approach to Analyzing the Software Process Change Impact Using Process Slicing and Simulation," *J. Syst. Softw.*, vol. 84, pp. 528–543, 2011.
- [24] Y. Ge and L. Bai, "Probability-based Safety Related Requirements Change Impact Analysis," *IEEE*, pp. 719 – 722, 2010.
- [25] H. Hindmarch, A. W. Gale, and R. E. Harrison, "A Support Tool for Assessing the Impact of Design Changes During Built Environment Projects," *IEEE*, pp. 1583–1587, 2010.
- [26] S. Nadi, R. Holt, and S. Mankovskii, "Does the past say it all? Using history to predict change sets in a CMDB," *IEEE 14th Eur. Conf. Softw. Maint. Reengineering*, pp. 97–106, 2010.
- [27] R. C. Lunardi, F. G. Andreis, W. Luis da Costa Cordeiro, and J. Araujo Wickboldt, "On Strategies for Planning the Assignment of Human Resources to IT Change Activities," *IEEE Netw. Oper. Manag. Symp.*, pp. 248–255, 2010.
- [28] M. Rahmouni, C. Bartolini, and H. P. Labs, "Learning from Past Experiences to Enhance Decision Support in IT Change Management," *IEEE*, pp. 408–415, 2010.
- [29] Z. Jiayi, L. Yunjuan, and G. Yuesheng, "The Requirements change Analysis for Different level users," *Int. Symp. Intell. Inf. Technol. Appl. Work.*, pp. 987–989, 2008.
- [30] J. F. Hoorn, E. A. Konijn, H. van Vliet, and G. van der Ver, "Requirement Change: Fear Dictate the Must Haves; Desires the Won't Have," *J. Def. Softw. Eng.*, vol. 80, pp. 328–355, 2007.
- [31] L. Wallace and M. Keil, "Software project risks and their effect on outcomes," *Commun. ACM*, vol. 47, no. 4, pp. 68–73, Apr. 2004.
- [32] R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying Software Project Risk: An International Delphi Study," *Jounal Manag. Inf. Syst.*, vol. 17, no. 4, pp. 5–36, 2001.
- [33] W.-M. Han and S.-J. Huang, "An empirical analysis of risk components and performance on software projects," *J. Syst. Softw.*, vol. 80, pp. 42–50, 2007.
- [34] B. B. Chua, "Rework Requirement Changes in Software Maintenance," *Int. Conf. Softw. Eng. Adv.*, pp. 252 – 258, 2010.
- [35] H. Kaiya, K. Hara, K. Kobayashi, A. Osada, and K. Kaijiri, "Exploring How to Support Software Revision in Software Non-intensive Projects Using Existing Techniques," in *IEEE 35th Annual Computer Software and Applications Conference Workshops*, 2011, pp. 327–334.
- [36] U. Vora, "Change Impact Analysis and Software Evolution Specification for Continually Evolving Systems," *2010 Fifth Int. Conf. Softw. Eng. Adv.*, pp. 238 – 243, 2010.
- [37] T. M. Pigoski, "Software Maintenance," in *The Guide to the Software Engineering Body of Knowledge (SWEBOK)*, no. May, 2001, pp. 1–16.
- [38] N. F. Schneidewind, "Predicting risk as a function of risk factors," *Innov. Syst. Softw. Eng.*, vol. 1, pp. 63–70, Mar. 2005.
- [39] N. F. Schneidewind, "Investigation of the Risk to Software Reliability and Maintainability of Requirements Changes," *Proceedings. IEEE Int. Conf. Softw. Maint.*, pp. 127–136, 2001.
- [40] N. Nurmuliani, D. Zowghi, and S. P. Williams, "Using card sorting technique to classify requirements change," *Proceedings. 12th IEEE Int. Requir. Eng. Conf. 2004.*, pp. 224–232, 2004.
- [41] R. Lagerstrom, L. M. von Wurtemberg, H. Holm, and O. Luczak, "Identifying factors affecting software development cost and productivity," *Softw. Qual. J.*, pp. 1 – 23, 2011.
- [42] R. Reboucas, J. Sauve, A. Moura, C. Bartolini, and D. Trastour, "A decision support tool to optimize scheduling of IT changes," *10th IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, pp. 343 – 352, 2007.
- [43] M. A. Rahman, R. Razali, and D. S. V. Singh, "Model Analisis Impak Perubahan Keperluan Perisian," *Asia-Pacific J. Inf. Technol. Multimed.*, vol. 3, no. 1, pp. 33–42, 2014.
- [44] S. Foo and A. Muruganatham, "Software risk assessment model," *IEEE*, pp. 536 – 544, 2000.
- [45] B. Alenljung and A. Persson, "Portraying the practice of decision-making in requirements engineering: a case of large scale bespoke development," *Requir. Eng.*, vol. 13, no. 4, pp. 257–279, Aug. 2008.
- [46] M. A. Rahman, R. Razali, and D. Singh, "A Conceptual Model of Impact Analysis for Requirements Change," *AWERProcedia Inf. Technol. Comput. Sci.*, vol. 03, pp. 763–770, 2013.
- [47] J. C. Anderson and D. W. Gerbing, "Structural Equation Modeling in Practice: A Review and Recommended Two-Step Approach," *Psychol. Bull.*, vol. 103, no. 3, pp. 411–423, 1988.
- [48] D. Gefen and D. Straub, "A Practical Guide To Factorial Validity Using PLS- Graph : Tutorial And Annotated Example," *Commun. ACM*, vol. 16, pp. 91–109, 2005.
- [49] J. Henseler, C. M. Ringle, and R. R. Sinkovics, "The Use Of Partial Least Squares Path Modelling In International Marketing," *Adv. Int. Mark.*, vol. 20, pp. 277–319, 2009.
- [50] W. W. Chin, *The Partial Least Squares approach to Structural*. London, UK: Erlbaum Associates, 1998.
- [51] J. F. Hair, G. T. M. Hult, C. M. Ringle, and M. Sarstedt, *A Primer on Partial Least Squares Structural Equation Modelling (PLS-SEM)*. Thousand Oaks: SAGE Publications, Inc, 2014.
- [52] J. C. Nunnally and I. H. Bernstein, *Psychometric theory*, 3rd Ed. New York: McGraw Hill, 1994.
- [53] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. New Jersey: Lawrence Erlbaum Associates, 1988.
- [54] N. Nurmuliani, D. Zowghi, and S. P. Williams, "Requirements Volatility and Its Impact on Change Effort : Evidence-based Research in Software Development Projects," in *AWRE 2006*, 2006.