THE EMERGENCE OF USER REQUIREMENT RISK IN INFORMATION SYSTEM DEVELOPMENT FOR INDUSTRY NEEDS

Fransiskus Adikara, Benhard Sitohang, Bayu Hendradjaya

School of Electrical Engineering & Informatics, Institute of Technology Bandung
JL Ganesha no 10, Bandung, Indonesia
E-mail: adikara@students.itb.ac.id,benhard@stei.itb.ac.id,bayu@stei.itb.ac.id

ABSTRACT

Many organizations require a more effective and powerful information system to support their daily business activities. To develop these systems, engineering process requirement needs and its respective risks must be properly. Gathering requirement risks are a challenging process as it involves stakeholders, users and developers. This paper reviews the current researches on risks assessment methods. We have analyze, characterized and classified these similar risks into knowledge, requirement volatility and documentation risks. We conclude this paper by discussing how the new risk classification can be applied to help organization in risk identification and finally, help reduce these risks.

Key words :*user requirements, requirements engineering, risk management, organization needs, information system development*

1. INTRODUCTION

Nowadays, the use of information is on the rise. "System" does not only hold an important role in the company, but has also become an added value that significantly increase the competitive advantage of the business. However, Information systems in Indonesia as well as other developing countries are still facing many development problems. Many of these systems fail to deliver and fulfill business needs of the companies.

Most information system development problems occur when requirement engineering process is not fulfilled. Requirement Engineering is a sub-area of Software Engineering that emphasizes on what a software must or must not do (Zave, 1997). The goal of requirements engineering is to provide the software engineer with methods, techniques and tools to help on the process of understanding and registering on what a software must do, so it can help stakeholder understands what to build before system development starts (Haron, 2010).

One category in requirements engineering is user requirements. User requirements is requirements process that capture user's tasks and business needs so that the analysts can determine ways in which the new systems can support user's needs and requirements. A good developer analyst teams are usually needed to elicit user's needs as user-related factors often poses a challenge obtaining and understanding their needs.

User requirement process often poses risks in system development process. Potential threat, vulnerabilities and uncertainties in user requirements process could create many problems in software development process. These risks are normally caused by interaction between stakeholder, user and developers team. We have identified some requirement risks that occur during requirements engineering process. In this paper, we shall focus more on the requirements risk that are most related to user requirements process.

Improper requirement engineering has significant impact in success rate of software development projects. Our research shows that problems in requirements engineering is one of the major cause of project over budget, delay, and scope reduction which significantly reduces the effectiveness of the final software to the business (Nancy, 2005). In worst case scenario, projects are

cancelled due to improper requirements engineering. Fixing errors in requirements engineering, when the projects are already at a later stage, process would elevate the costs in requirements engineering to be 100 times expensive than other stages of the projects, such as coding (Boehm, 1983). As a result, the cost in requirement engineering could reach up to 40% to 50% of total project costs (Nancy, 2005). It is therefore crucial for industries and companies to understand the importance of this phase in ensuring the success of system development.

Main aim of this paper is to show the emergence of risk in information system development caused by user requirements process and re-categorize the risks involved, to increase industry's awareness of the risk factors in system development. This paper consists of 5 sections: section 2 describes the literature review, and prior research of requirement engineering risk, section 3 describes research methodology, section 4 defines the risk in user requirements and finally in section 5 draws some conclusions and discusses future work.

2. RESEARCH IN REQUIREMENTS RISK

Requirements consist of five categories: business requirements, user requirements, functional requirements, non-functional requirements, and system requirements (Dennis, 2012). User requirements is one of requirements categories that define and analyze user's needs, tasks, activities, and process to achieve business objectives (Dennis, 2012). User requirements are mostly used by user-center software engineering process and most of literature points out this requirements as an important phase and category in requirements engineering (ISO, 1999). Simple method of user requirements process shown in Figure 1.

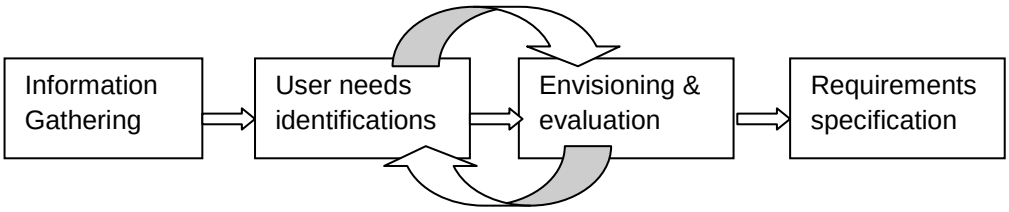

Figure 1. General Process for user requirements analysis (Maguire, 2002)

Many elicitation techniques can used for eliciting user needs. The elicitation process will be better if process, methods, and capture mechanism are well defined, documented, and communicated to the related stakeholders since the beginning of the elicitation process (Berenbach, 2009). User requirements must be realistic, which means user requirements must be clear, verifiable, complete, accurate, and feasible (BSSC, 1995).

In prior research, requirements risk have been divided into three types, such as changing requirements risk, misunderstanding requirements risk, and incomplete requirements risk (Parinyavuttichai, 2011). Another risk mentioned is cultural differentiation between users and developers which causes difficulty in requirements engineering (Boehm, 1983). Requirements risk also occurs when users and stakeholder have lack of knowledge in education about software engineering (Armarego, 2005).

In requirements process, mostly developers and user do not use storing media or tools that can keep all the progress in requirements process (Nancy, 2005). Other research discusses security requirements and legal requirements. Security requirements are usually specified to prevent any activities that may pose a threat to either the stakeholders or the system itself (Markose, 2008). Furthermore, in legal and regulation dimension, legal requirements also occurs to make sure the systems don't break the regulations in the process or output (Compagna, 2012).

3. RESEARCH METHODOLOGY

To reach our aim in identifying and re-categorizing user requirements risks in system development, our first step to research literature, journal or proceeding with requirement engineering, requirements risk, and user requirements as key words. After collecting literature, we select the literature that focuses on requirements engineering risk and is related user requirements problems. Lastly, we summarize the problems and risks put forward in the literature. Next, we categorize them into user requirements by industry so that industry could take notice and be aware of all the risks that can affect requirement process in developing a new system.

4. RISK IN INFORMATION SYSTEM DEVELOPMENT PROCESS THAT RELATED WITH USER REQUIREMENTS PROCESS

As previously stated above, threat, vulnerabilities, and uncertainties in user requirements process can create many problems in further software development process (Cheng, 2007).

Most of our findings from current literature and study, discusses problems in information system development process, but not specifically on requirement risks especially on user requirements risk. Risks caused by user requirements must be detected so industry can manage the risk when developing new software or systems to gain competitive advantage.

Contrary to common belief, our research shows that what is important in determining the success of a project is not the number of users involved, but rather the size of the project in terms of functionality (Verner, 2006a). Further, research shows that effective Project Managers are those who could manage requirements effectively, not just managers who have experienced in the application area (Verner, 2006b).

To help industry undergo right and proper requirement engineering process, they need to focus on all risks that are most relevant to their business process. Therefore, a classifications of risks in user requirement engineering is very beneficial to aid industry in identifying their risks. In the next paragraph, we will focus on defining and categorizing numerous user requirement process risks into 3 classifications, namely: knowledge risk, requirements volatility risk and documentation risk.

4.1. Knowledge Risk

Knowledge risk consists of risk and problems that occur in information system development. This is because stakeholder, users, or developers have lack of knowledge in software engineering process, especially in requirements engineering methodology, models, and techniques, and also on supportive and non-functional requirements information that needed. Stakeholders, users, and developers teams have to leverage and align their knowledge on software engineering and requirements engineering to develop a new system.

Both waterfall model and Total Quality Management “voice of the customer” guideline states that users' needs must be recorded to determine the function of the new systems. Problems usually arise if the users have no idea of the relative cost and difficulty of requirements (Boehm, 1999). This problem often occurs because of user's lack of knowledge on requirements process and thus, they are unable to calculate cost of the process.

Another risk that is related to knowledge risk in user requirements is users' lack of education in software engineering (Armarego, 2005). Gap between education and industry needs in software engineering domains is still large (Training, 2007),(Mann, 2002). Lack of interest in requirement engineering course and difficulty in learning requirements modeling and analysis, requirements management and requirements documentations activities result in insufficient competency in requirement process among students (Memon, 2010). Thus, when these students take the roles of users or developers team to develop a new systems, they face many problems in requirements elicitations process.

Risks in non-functional requirements consists of possible security and legal requirements risk. Security requirements are usually specified to prevent any activities that may pose a threat to either the stakeholders or the system itself (Markose, 2008). These risks may emerge due to users' and developers' lack of knowledge in security domain, resulting in inadequate system security strategy. Thus systems can be accessed by unauthorized users.

Inadequate understanding and knowledge on legal and relevant other legal regulations is another knowledge risk that is common among users and developers. An example of such risk is insufficient knowledge on tax rules that will affect tax calculation requirements in the system being developed (Compagna, 2012).

4.2. Requirements Volatility Risk

Requirements volatility refers to additions, deletions and modifications of requirements during the systems development process (Sachidanandam, 2010). Requirements volatility risk is a problem in user requirements process that occur when the requirements change because of relation between stakeholder, users and developers.

Requirements volatility risk is situation when system developers are confused by uncertainties due to constantly changing requirements risk that could lead to development costs and schedule changes. Change of requirements usually occur due to major addition or modification to the systems requirements which arises because of changes in in technological or business process, conflict between departments, lack of understanding of the systems development among users, and also choice of software engineering methodology (Parinyavuttichai, 2011). Requirement changes can be categorizes as external change, internal change, technical change, and learning process (Perumal, 2011).

Misunderstanding requirements risk is another common volatility risk. This risk is a condition where users and developers team have different expectation of information systems as a result of little understanding in the systems requirements. Communication between users and systems developers is the main issues in this situation as miscommunication can lead to different point of view about the new systems (Parinyavuttichai, 2011).

Incomplete requirements risk is a situation where some user requirements are ignored or overlooked by the project team or users themselves (Lauesen, 2001). Incomplete requirements could cause defect in requirement process (Lawrence, 2001). For instance, users may change their expectations of an information system over time and the developers did not choose an appropriate methodology to elicit user requirements and evaluate or confirm this methodology against end-user expectations (Parinyavuttichai, 2011); Another factor that could create incomplete requirements risks is when not all stakeholder participate in requirements process (Nancy, 2005). Incompleteness can also occurs when some of requirements are hidden and not easily collected by developers. This incomplete requirements risk also can be classified as requirements volatility risk.

Based on the above explanation, it is very important for users and developers to gain the same point of view on the current systems and future systems being developed. Users and developers should align their point of view with the right techniques and methods. Communication is the key factor in this process to bridge the mindset between users and developers. Requirement volatility risk emerges because a lack of communication, resulting in different point of views between users and developers.

4.3. Documentation Risk

Documentation risk occurs when developers and users do not use proper storage media and tools to document the progress in requirements process (Nancy, 2005). This results in loss of attributes, difficulty in tracing, scheduling and prioritizing of tasks in the requirements process activities (Nancy, 2005). Proper documentation plays an important role in the situation when user wishes to change or remove some requirements.It would also hinder knowledge transfer in the case where users and

developers have moved out from the projects.

Documentation risk also occurs when users do not understand the documentation model used by developers and thus user is unable to check the process in the future (Lawrence, 2001).

4.4. Discussion

In conclusion, we can foresee that there are factors that need to be fixed and improved to fulfill proper requirements and minimize the risk caused by user requirements process.

Future works on knowledge risk should make further assessment methodology and create competency leveling for users, developers and stakeholders as pre-requirements that must be done before industry decide to create or develop a new system. Knowledge in software engineering and requirements engineering, must be aligned with industry needs so developers can have more effective solution with the new system developed. Users also have to leverage their knowledge in system development process to support and give suggestion to new information system design. In addition, stakeholder, who initiate and lead the system projects, needs to understand and have proper knowledge on software engineering especially requirements engineering.

Industry should also be more aware about volatility risk, to minimize or eliminate factors that could change and pose as obstacle in the development process. Understanding volatility risk is important so that the industry can be more selective in choosing proper requirements elicitation methodology and modeling techniques.

Lastly, we need to design and choose right tools and media storage to eliminate documentation risk. A good knowledge management system is needed especially for requirements documents process.

With this defined categorization, the industry should be able to easily identify main problems in users requirements process during new systems development process.

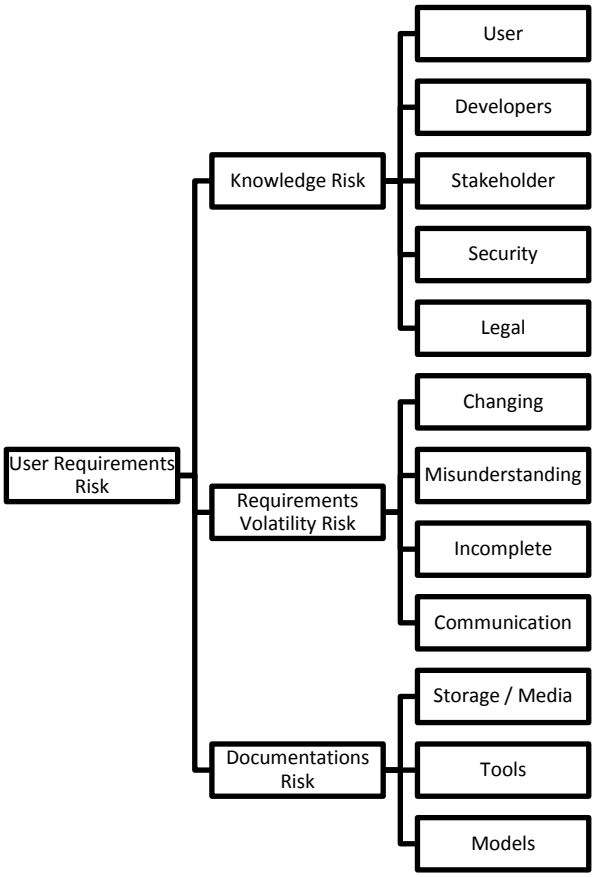

Figure 2. User Requirements Risk Classification

5. CONCLUSION AND FUTURE WORK

From this paper, we can see the emergence of risk in information system that caused by user requirements process. From those facts we can conclude that there are still a lot of problems we must learn and manage in user requirements process.

The risk caused by user requirements process can be categorized in three major categories: knowledge risk, the volatility risk, and documentations risk. This categorization is based on key problems found in user requirements process. Further research is needed to know how to manage and eliminate this risk so we could deliver better quality of systems.

As previously mentioned, prior research concludes that the size of the project in terms of functionalities is more important tan user requirements. In addition, information system needs a Project Manager who could manage requirements effectively, not just an

experienced Project Manager. This conclusion encourages us to further study on key factors in requirements process. Does user really have a big effect on system development?

From this paper, we also found that one of the main problems in the information system development is human resources in stakeholder position or developers teams, but it also contradictive with the literature review that still put human resource as main factors in successful Requirements Engineering (Dennis, 2012), (Miller, 2008).

New problems and the emergence of risk in information system development due to user requirements process should be solved. it is urgent for us to develop methods and requirement engineering framework that can eliminate or minimize all these risks. We also need to find solution to reduce the gap between education/theories and industrial/practical needs so we can increase the quality of information systems development.

6. REFERENCES

(a) (BSSC), (1995). ESA Board for Software Standardisation and Control. *Guide to the User Requirements Definition Phase.* Paris: European Space Agency.
(b) Armarego, J., and O. Minor. (2005). "Studio Learning of Requirments: Towards aligning Teaching to Practitioner Needs." RETT.
(c) Berenbach, Brian, Daniel J. Paulish, Juergen Kazmeier, and Arnold Rudorfer. (2009). *Software & Systems Requirements Engineering: In Practice.* McGraw Hill.
(d) Boehm, B.W. (1983). "The Economics of Software Maintance." Software Maintance Workshop.
(e) Boehm, Barry, D. Port Marwan Abo-Antoun, J. Kwan, and A. Linch. (1999). "Requirements Engineering, Expectations Management, and the Two Cultures." The 4th IEEE International Symposium on Requirements Engineering.
(f) Cheng, Betty H.C., and Joane M. Attle. (2007). "Research Direction in Requirements Engineering." (FOSE).
(g) Compagna, L., P. El Khoury, A. Krausova, F. Massacci, and N. Zannone. "How to Integrate Legal Requirements into A Requirements Engineering Methodology for the Development of Security and Privacy Patterns." 2008. http://security1.win.tue.nl/~zannone/publication/comp-elkh-krau-mass-zann-08-AIL.pdf (accessed 11 22, 2012).
(h) Dennis, Alan, Barbara Haley Wixom, and Roberta M. Roth. (2012). *System Analysis and Design: 5th Edition.* Wiley.
(i) Haron, Azlena, and Sahsul Sahibuddin. (2010). "The Strength and Weakness of Requirement Engineering (RE) Process." 2nd ICCTD.
(j) ISO. (1999). *ISO 13407: Human-centred design processes for interactive systems.* International Standards Organisation.
(k) Lauesen, S., and O. Vinter. (2001). "Preventing Requirement Defects: An Experiment in Process Improvement." (Requirements Engineering 6).
(l) Lawrence, Brian, Karl Wiegers, and Christof Ebert. (2001). "The Top Risks of Requirements Engineering." In *IEEE SOFTWARE*. IEEE.
(m) Maguire, Martin, and Nigel Bevan. (2002). "User requirements analysis: A review of supporting methods." Montreal: Proceedings of IFIP 17th World Computer Congress.
(n) Mann, Joan. (2002). "IT Education's Failure to Deliver Successful Information Systems: Now is the Time to Address the IT-User Gap." (Journal of Information Technology Education) 1, no. 4.
(o) Markose, Sojan, Xiaoqing (Frank) Liu, and Bruce McMillin. (2008). "A Systematic Framework for Structured Object-Oriented Security Requirements Analysis in Embedded Systems." IEEE/IFIP International Conference on Embedded and Ubiquitous Computing.
(p) Memon, Rafia Naz, Rodina Ahmad, and Siti Salwah Salim. (2010). "Problems in Requirements Engineering Education: A Survey." ACM, n.d.
(q) Miller, Granville, and Laurie Williams. (2008). "Personas: Moving Beyond Role-Based Requirements Engineering.".

(r) Nancy R. Mead, Ted Stehney. (2005). "Security Quality Requirements Engineering (SQUARE) Methodology." SESS 2005.
(s) Parinyavuttichai, Nipon, and Angela Lin. (2011). "Managing User Requirement Risks – An Exploratory Study of IS Projects from the Views of Outsourcing Teams." International Conference on Information Resources Management (CONF-IRM).
(t) Perumal, S. Arumuga, and G.Kavitha. (2011). "Changing Requirements – Correlated to Risk or." (IACSIT International Journal of Engineering and Technology) 3, no. 1.
(u) Sachidanandam Sakthivel. (2010). "Manage Requirements Volatility to Manage Risks in IS Development Projects." (ISACA Journal) 5.
(v) Training, M., and C. La. (2007). "What Are Employers Really Looking For ?" REET.
(w) Verner, June, Karl Cox, and S.J. Bleistein. (2006a). "Predicting Good Requirements for In-house Development Projects." ISESE'06 ACM.
(x) Verner, June, Karl Cox, Steven Bleistein, and Narciso Cerpa. (2006b). "Requirements Engineering and Software Project Success: An Industrial Survey in Australia and the U.S"
(y) Zave, Pamela, and Michael Jackson (1997). "Four Dark Corners of RE." *ACM Transaction on SE and Methodology*.

AUTHOR BIOGRAPHIES

Fransiskus Adikara is a doctorate candidate in software engineering, School of Electrical Engineering & Informatics, Institute of Technology Bandung, Bandung. He received his Bachelor Degree in Informatics in 2001 and Magister Management in Information System in 2003, both from Bina Nusantara University. His research interest are in the area of software engineering, balanced scorecards, and IT Governance. His email address is (adikara@students.itb.ac.id)

Benhard Sitohang is a Professor in Database System, and researcher at Software Engineering and Data Research Group, at School of Electrical Engineering and Informatics, Institute of Technology Bandung. As Electrical engineering (1978), Received Master Degree (1980) and Doctor degree in Database System (1983) from USTL – France.

Bayu Hendradjaya is a lecturer in software engineering and a researcher at Software Engineering and Data Research Group, at School of Electrical Engineering and Informatics, Institute of Technology Bandung (ITB). He received a Bachelor Degree (1994) in Informatics, a Master Degree with honors (1997) in Software Engineering both from ITB and PhD in Software Engineering (2010) from La Trobe University. He has been working as a software consultant and IT auditor in many government institutions and banking in Indonesia. He also has been working as a senior software analyst and developer in two international companies. He has broad interests in software measurement and metrics, software testing, requirement engineering, software process improvement, component based software engineering and Agile software development methodology.