RISK MANAGEMENT METHOD IN IT PROJECT: A REVIEW

Ikhtiar Faahakhododo[1]; Indra Dwi Rianto[2]

[1,2]Computer Science Department, School of Computer Science, Bina Nusantara University
Jln. K.H. Syahdan No. 9 Palmerah, Jakarta Barat, 11480
[1]ifaahakhododo@binus.edu; [2]indra.rianto@binus.edu

ABSTRACT

In the development of a software, there are several aspects that must be taken to ensure that the process can produce a useful product and make a profit. This article clarified some of the methods of risk management exist. There was two techniques to determine the risks used in this study, those were Metrics of Process Structure and Referential Model or could be referred as the Comparison to the Referential Model technique. That technique will produce Software Process Meta Model, Model of Risk Management, and Manage Risks in Project models. Those models were used to help managers in mapping the risks of the project.

Keywords: *risk management, project, IT project*

INTRODUCTION

Software engineering is a discipline or technique that deals with all aspects of software production range from an early stage that the system specification to system maintenance has even begun to use by the user. With so many systems that are controlled using software, the dependence on software is increasing. The ability of the system to produce high quality, reliable, reliable, fast, and economical is a demand that must be answered by professional developers. Software engineering is not only concerned with the theory and methods but also as a tool for software development professionals.

Software development project is a high-risk activity, complicated, and requires more effort to achieve its goal (Keshlaf & Riddle, 2010). The risk of software development increased along with the development of the software industry. Many software development projects that do not achieve their goal to produce software that can be accepted within the limits that have been agreed, such as time, budget and quality for their risk and the absence of Risk Management.

Risk, in the area of software, has represented systematically by Boehm in the 80s, through the spiral model. This model has fundamental to be repeated and risk analysis in each iteration. From this view, to achieve success requires more than a good process and intuitive thinking ability, which requires discipline. This discipline is called risk management. Currently, the management of risk in software engineering is an evolution of a concept that evolved from the risk management process model analysis, which should include all processes in the software life cycle. Risks can't be as simple in the project, but the risk should be the core business. Risk management also has a proactive focus on preventing a problem that has not yet appeared, but can occur continuously and simultaneously (Dhlamini, Nhamu, & Kachepa, 2009).

There are many risks contained in software development that are high quality and within budget. But it is good to take this risk, since it must be accompanied by comparable awards. The larger the risk, the greater the reward will be given. In software development, award/reward obtained can be

high, but goes straight to the disaster/damage that can occur in the development of the software. Need of software risk management is illustrated in Gilb risk principle is: "If we are not actively attack risk, then the risk will be active against us" (Westfall, 2001). All projects share a certain threshold of risk, and most projects of Information Technology (IT) has a considerable risk. Risk can be reduced, organized, and managed in accordance with rigorous planning and assessment (Sharif & Basri, 2011).

The purpose of this study is to give the alternative way to map and manage the risks that might appeared either from the internal or external of an IT project. Thus, it is hoped that this study could help any project managers in the future to provide more attention to risk management and help them to map the risk that could they face in software development project so they could prepare a strategy once they face risks in their IT project.

METHODS

Research methodology that used in this article are categorized as a qualitative method, involved: (1) Data Collection: All the data are collected from study literature, collect any data and information from several sources like textbooks and Journal, (2) Analysis Method: In this paper, analysis method using Metrics of Process Structure technique and comparison to referential model, and (3) Conclusion.

RESULTS AND DISCUSSIONS

There are several ways to determine the risks that will be encountered in the development process. Some of the ways are: (1) Metrics of Process Structure and (2) Comparison of the Referential Model. The first technique is using metrics of process structure to focus on the most important element and the most at risk of a process. The second technique focuses on the differences between actual and referential models (Miler & Górski, 2004).

In the metrics of process structure technique, it figures out a way it metrics, which element is the most important and the most at risk of the process to be further investigated by a number of general questions. The procedure of risk identification involves two stages: (1) counting metric models and (2) studying the context of the process elements, as seen in Figure 1.

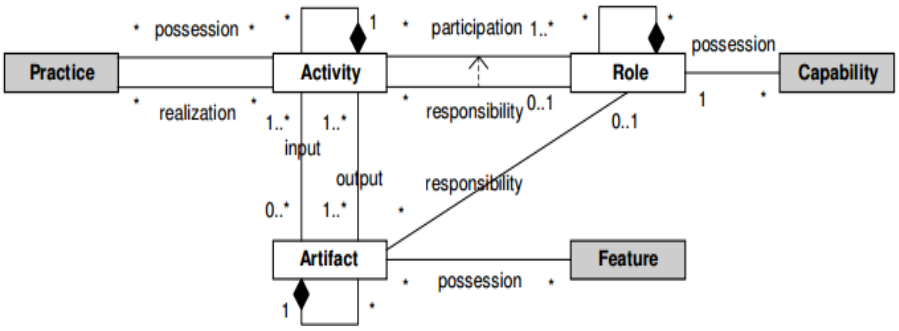

Figure 1 Software Process Meta Model

Figure 1 is an example of a software process meta-model. From this image, we can calculate the metric models by defining a set as follows: (1) $A_{in}(Ar)$ – set of activities that have artifact Ar as input. (2) $A_{out}(Ar)$ – set of activities that have artifact Ar as output. (3) $A_{part}(R)$ – set of activities in which role R participates. (4) $Ar_{in}(A)$ – set of input artifacts of activity A. (5) $Ar_{out}(A)$ – set of output artifacts of activity A. (6) $R_{part}(A)$ – set of roles participating in activity A. (7) P(A) – set of practices possessed by activity A. (8) F(Ar) – set of features possessed by artifact Ar. (9) C(R) – set of capabilities possessed by role R.

The following metric elements are calculated to determine which model should be identified more and focus. In the following definition, symbol |A| shows the cardinality of set A.

I (A) – significance of activity A
Total output level of the importance of all the artifacts of activity A

$$I(A) = \sum_{Ar \in Arout(A)} I(Ar), I(A) \in N \tag{1}$$

The average rate of importance on the output artifact of the activity A

$$\overline{I(A)} = \frac{I(A)}{|Arout(A)|}, \overline{I(A)} \in R + \cup \{0\} \tag{2}$$

The number of input artifacts of activity A

$$I(A) = |Arin\ (A)|, I(A) \in N \tag{3}$$

I(Ar) – importance rate of artifact Ar
The number of activity of the input artifact Ar

$$I(Ar) = |Ain\ (Ar)|, I(Ar) \in N \tag{4}$$

I(R) – importance rate of role R
The number of role R

$$I(R) = |Apart\ (R)|, I(R) \in N \tag{5}$$

R(A) – risk of activity A
Importance rate of activity A divided by the amount of activity A

$$R(A) = \frac{I(A)}{|P(A)|}, R(A) \in R + \cup \{0\} \tag{6}$$

Rin(Ar) – risk of input artifact Ar
Importance rate of artifact Ar divided by the number of features of the artifact Ar

$$Rin(Ar) = \frac{I(Ar)}{|F(Ar)|}, Rin(Ar) \in R + \cup \{0\} \tag{7}$$

The risk of developing the artifact Ar with Activity A is the number of feature of artifact A divided by the total number of the features of all the input artifacts of activity A plus the amount of

involvement of the activity A plus the total amount of the capabilities of all role that participating in activities.

$$Rout(Ar,A) = \frac{|F(Ar)|}{\sum_{Ar' \in Arin(A)}|F(Ar')| + |P(A)| + \sum_{R \in Rpart(A)}|C(R)|}, Rout(Ar,A) \in R+ \cup \{0\} \quad (8)$$

The total risk of the development of the artifact Ar concluded for all the activities that have the artifact Ar as output

$$Rout(Ar) = \sum_{A \in Arout(Ar)} Rout(Ar,A), Rout(Ar) \in R+ \cup \{0\} \quad (9)$$

The average of the artifact Ar development is a risk average of the development of artifact Ar for all the activities that have the artifact Ar as the output.

$$\overline{Rout(Ar)} = \frac{Rout(Ar)}{|Aout(Ar)|}, \overline{Rout(Ar)} \in R+ \cup \{0\} \quad (10)$$

R(R) – risk of role R
The importance rate of the role of R divided by the numbers of the capabilities of the role R

$$R(R) = \frac{I(R)}{|C(R)|}, R(R) \in R+ \cup \{0\} \quad (11)$$

After calculating the metric models, the next step is to study the context of process elements. For the riskiest element models (according to the metric calculated in the previous step) these following questions are about the context: (1) Do you have the ability and the input artifact to perform the activities? (2) Do you have an activity, practice or ability to obtain input artifact? (3) Do you have any feedback output artifact to artifact? (4) Do you have the practice or the ability to develop the production of artifacts? (5) Do you have a practice and/or the ability to build features into the output artifact? (6) Do you have the ability and input artifacts to make practice? (7) Do you have activities, practices and/or the ability to create a role? (8) Do you have a practice and/or the ability to build capability in the role? (9) Do you have a role that is responsible for artifacts? (10) Do you have a role that is responsible for the activity? (11) Do you have any artifacts (e.g., guidelines, standards, measures, literature) that define activities, artifacts, roles, practices, features, capabilities?

A negative answer indicates an increase in risk associated with particular element models and showed a risk factor for the entire project. To clearly define the context of risk, the risk factors can be expressed by risk patterns.

There is another technique besides metrics of process structure, the technique is Comparison to Referential Model or could be called the Referential Model Comparison. The technique is based on a comparison of a model element that analyzed for referential semantic models. The result is a list of the missing elements and the factors that show excessive risks that might occur. These are the two stages in comparison techniques referential models: (1) mapping the model that was analyzed by referential models: this step involves mapping together with similar semantic elements both models. In general, such a joint mapping is a many-to-many relationship. It is also limited by the assumption that the elements are mapped to be of the same type (an activity can be mapped only on activities, artifacts and so on). (2) Find the differences between models: at this stage, all the elements and relationships are analyzed, and referential model examined.

The identified differences are collected in the following: (1) missing list of activities, artifacts, roles, practices, features and capabilities, (2) a list of the relationship between missing model elements, (3) excessive list of activities, artifacts, roles, practices, features and capabilities, (4) a list of a redundant relationship between model elements. The differences in the list indicate a potential risk factor for the project. Analog with risk identification technique based on metrics, risk context, can be specified with the pattern of risk.

In addition to these two techniques, there are other ways to identify risks; these are five steps to identify the risks: (1) selection of the target values that threatened by the risk. (2) Identify the context of the use of systems, services, scenarios and use cases. (3) Build a failure mode map with additional questions. (4) Identify the real risks. (5) Review and documentation of the identified risks. (Miler, 2005)

In Software Risk Management Process there are several models available for it. Figure 2 shows a model as the result of the development of the Software Engineering Institute (SEI) (Chowdhury & Arefeen, 2011).

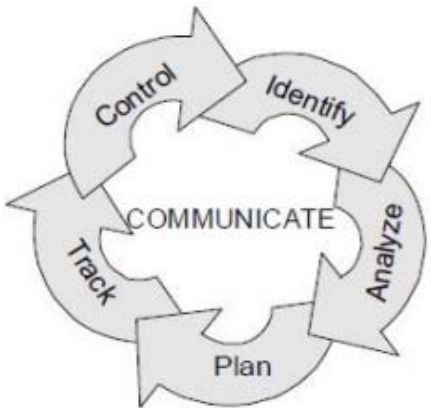

Figure 2 Model of Risk Management

First, identify. Before a risk can be controlled/regulated, these risks must be identified before giving negative effects to the project. Forming an environment that encourages people to voice concerns and issues – issues and review the quality throughout all phases of the project is a common technique for identifying risks.

Second, analyze. The analysis is a conversion of risk data to be a risk-informed decision-making. This includes reviewing, prioritizing, and selecting the most important risks to overcome. The Software Risk Evaluation (SRE) Team analyzes each identified risk in terms of the impact on cost, schedule, performance, and quality of products.

Third, plan. Planning change risk information into decisions and actions for the present and future. Planning involves the development of action to individual risks, prioritizing actions and the risk of making the Risk Management Plan. The key to risk action plan is to consider the future consequences of decisions made today.

Fourth, track**.** Searching consists of monitoring the status of risk and actions taken to mitigate against risks.

Fifth, control. Depending on the risk control project management process to control the risk action plan, improving the variation of the plan, responded activities that trigger risk, and improve the risk management process.

Sixth, communicate. Communication occurs throughout all of the risk management function. Without effective communication, no risk management approaches that could be eligible. It is an integral part of all other risk management activities.

Every asset has a cost that each - each. The cost of physical assets should cover the rate of inflation, at least the same as the cost of replacement. Here are some categories to be considered (Meritt, 1998). First, facility**:** all buildings, air conditioners, furniture and other ancillary equipment. Think of the things like "fire" or "flood." Other possibilities include earthquakes, bombs and chemical contamination, which causes EPA. Second, equipment: all the information system equipment is placed in the adjacent area. Not including equipment that will not be lost, for example, a fire that completely destroyed the computer facilities.

Third, software: all programs and documentation would be lost if the computer facilities were destroyed. It can be split into two (1) commercial: this software is not free, and can be consulted if an error occurs. Check the warranty form, because it can be replaced free of charge in the event of disaster. (2) Proprietary: this software is an in-house software. How much is it if the software is remade.

Fourth, record and files: all magnetic media data file will be lost if the facilities were completely destroyed. Just count and multiply. The information content of these items will be covered in the next. Fifth, data and information: value is methodically applied randomly to represent the value of all the data and the information maintained in computer facilities; including losses that may occur is compromised data but does not have to be destroyed.

Several techniques have been described in the previous section, but in reality, there are many software development projects that do not care about risk management because still not used and considers risk management simply discard - waste of time, effort, cost, and mind.

However, for large-scale projects that should pay attention to risk management, due to the cost needed in a large project is not small. When described, to manage risks in a project, can be described as Figure 3.

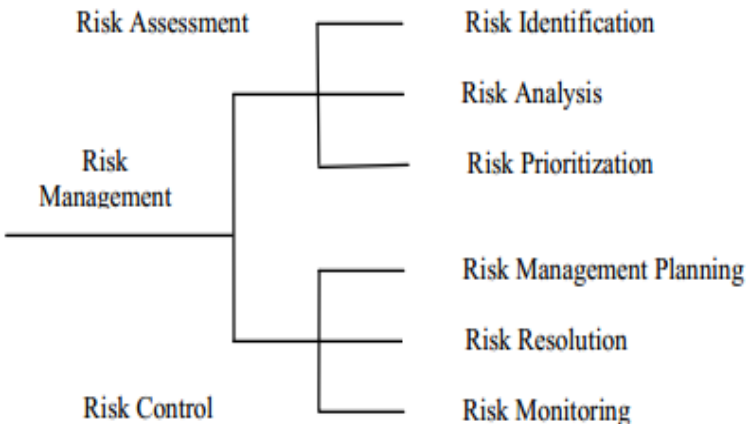

Figure 3 Manage Risk in Project 1

Figure 4 indicates two major steps, Risk Assessment, and Risk Control. Risk Assessment is more about mapping all the potential risks that could happen in the project. The subtasks from Risk Assessment start from Risk Identification followed by Risk Analysis and then we sort the risks by its priorities in Risk Prioritization subtask. The second major step is Risk Control Step, which is more about how we handle it.

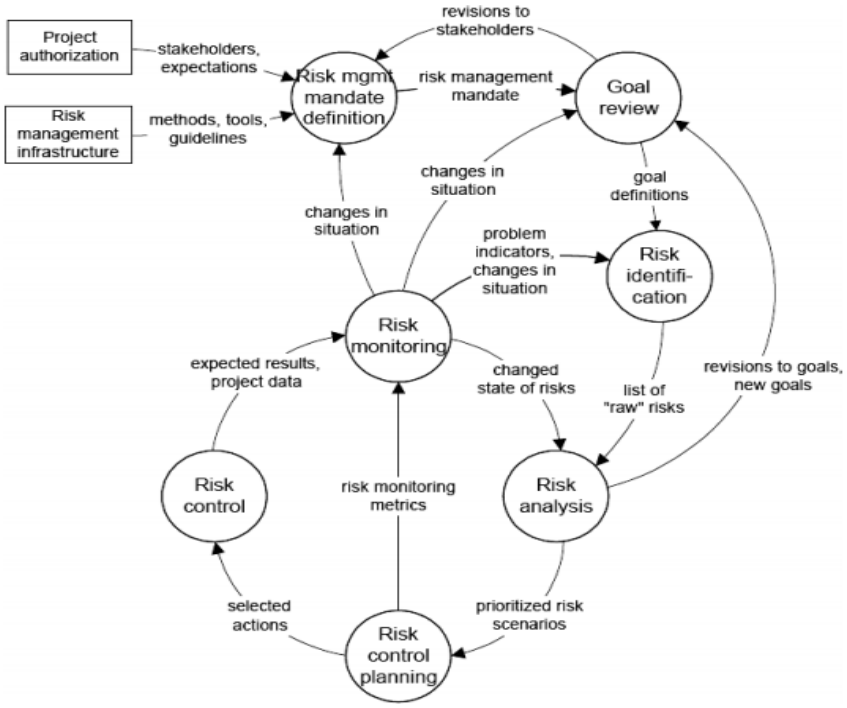

Figure 4 Manage Risk in Project 2

Figure 5 is about the flow of risks in a project and how it connects to the goal of the project. All the process is connected to each other.

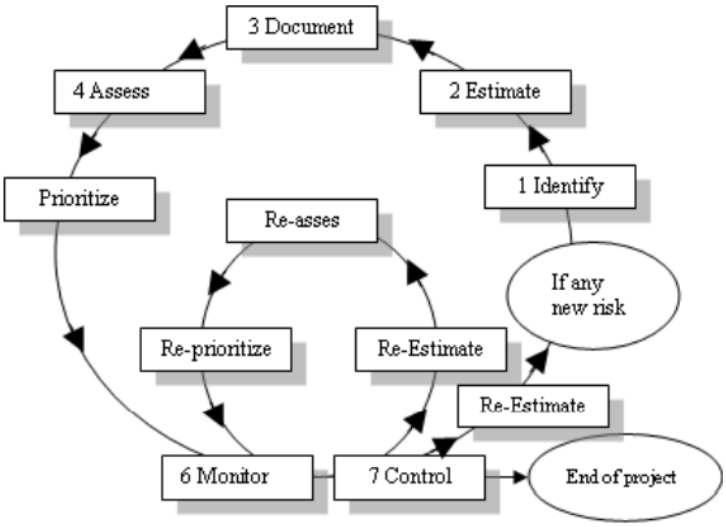

Figure 5 Manage Risk in Project 3

From the figure above, it shows that once a risk is identified, the risk is not only ignored but is controlled and monitored, because if it is left alone, it will not provide any results on a project.

In the literature, there are several methods that have been given, one of which is the metric. This method will be more beneficial when used in a large project, not a small project because it will be quite time-consuming in the calculation of the metrics process. While the techniques that can be acceptable to all projects is a technique that uses a model of the development of the Software Engineering Institute (SEI), in the method, the visible repetition of stages happen continuously to minimize the risk.

CONCLUSIONS

The number of unsuccessful projects turned out to be mostly caused by a lack of awareness of risk management. A lack of knowledge is also one of the reasons a strong supporter of a project is not successful. The number of stages in the identification process also makes most people hesitate to follow it as it makes the cost of an enlarged project. Therefore, all the decisions are determined on the project manager who leads the project. From this study, the researchers can use those techniques of the project risk management to mapping the risk of the project and helps us to identify the problems before it happens, so it could help the manager to plan a strategy as prevention.

REFERENCES

Chowdhury, A. A., & Arefeen, S. (2011). Software Risk Management: Importance and Practices. *JCIT*, *2*(1), 49-54.

Dhlamini, J., Nhamu, I., & Kachepa, A. (2009). *Intelligent Risk Management Tools for Software Development*, SACLA '09, 29 June - 1 July, Mpekweni Beach Resort, South Africa, 33-40. Retrieved from http://web.nchu.edu.tw/pweb/users/arborfish/lesson/8614.pdf.

Keshlaf, A. A., & Riddle, S. (2010). *Risk management for web and distributed software development projects*. Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference, Barcelona, 22-28.

Meritt, J. W. (1998). *Risk Management*. Retrieved from http://csrc.nist.gov/nissc/1998/proceedings/paperE5.pdf.

Miler, J. (2005). *A Service-Oriented Approach to the Identification of it RISK*. proc. of 1st IEEE International Conference on Technologies for Homeland Security and Safety TEHOSS'2005 September 28-30, Gdansk, Poland, 2-4.

Miler, J., & Górski, J. (2004). *Identifying Software Project Risks*. proc. of 17th International Conference "Software & Systems Engineering and their Applications" November 30 - December 2, Paris, France, 3.

Sharif, A. M., & Basri, S. (2011). A Study on Risk Assessment for Small and Medium Software Development Projects. *International Journal on New Computer Architectures and Their Applications (IJNCAA)*, *1*(2), 325-335.

Westfall, L. (2001). *Software Risk Management*. Retrieved from http://westfallteam.com/Papers/risk_management_paper.pdf