A Review Study: Automated Risk Identification Tool for Software Development Process

OMNITHAKUR* and JANPREET SINGH

Department of Computer Science Engineering, Lovely Professional University (Punjab) - 144 402, India.

(Received: February 19, 2014; Accepted: February 26, 2014)

ABSTRACT

Risk Identification tools are usually established for avoiding or minimizing problems, likely to occur during software development. It can be stated as the task of analysing and managing the impact of every important risk occurred in the project. In the context of Risk Identification tool practices, we developed a scoping study, aiming at analysing the current scenario of Risk Identification practices in software development process. We analysed different studies published by the most important venues published up to the year 2013. Based on the analysed data set, we sketched a set of useful techniques and tools for applying Risk Identification in software projects. The analysis indicates that most of the studies subjectively describe ways to evaluate risks, instead of providing readers with details on how Risk Identification is to be performed. Such findings points out to the need of further research in the field of Risk Management, especially for the identification of risks in software development process for better results.

Key words: Risk Management, Risk Identification, Risk Index, Risk Analysis, Risk Assessment.

INTRODUCTION

The risk management is all about managing the risks which occur. Many concepts of software risk management are there but the most important from them are risk index (can be categorised as high level risk, medium level risk and low level risk), risk assessment (it involves the accurate understanding of the current system and it’s all other features like security), risk analysis (used to identify the risk elements of a project with high probability). (Luanna Lopes Lobato, 2012)

As the technology is increasing very fast, the developed applications are having. The complex user interface and these developed applications run on the various systems simultaneous like applications which support client server architecture. Due to the complexity and the nature of the developed application these are able to run on the various operating systems. To overcome these types of problems we want to evaluate the performance and other factor of the application and this can be done by defining some set of rules. These types of problems include the
concept, strategies and practices of software engineering. With the help of including these concepts we can evaluate the performance and other relevant factors of the application. Before proceeding further we have to take in consideration some of the major failures that will lead to software failure before delivering the application to the customer. Those major failures:
1) It can be poor user input.
2) Stockholder Conflicts can be there.
3) Vague requirements.
4) It can be poor cost and schedule requirements.
5) Can be due to wrong analysis of the requirements.
6) Can be due to communication breakdown.
7) Can be due to architectural breakdown.

Risk management process
There are several models available for risk management. The model recommended in this section was developed by the Software Engineering Institute (SEI).

Plan
Planning turns risk information into decisions and actions for both the present and future. Planning involves developing actions to address individual risks, prioritizing risk actions and creating a Risk Management Plan. The key to risk action planning is to consider the future consequences of a decision made today.

Track
Tracking consists of monitoring the status of risks and the actions taken against risks to mitigate them.

Control
Risk control relies on project management processes to control risk action plans, correct for variations from plans, respond to triggering events, and improve risk.

The risk management process consists of ten steps as described in the diagram. Use of the activities associated with these steps constitutes an acceptable risk management approach and could be incorporated into a Risk Management Plan. The size, visibility, or consequences of the project drives the complexity of the process. The steps can be:
1) Identify Risks
2) Analyse Risk
3) Prioritize Risk
4) Identify Risk Aversion Methods
5) Identify Risk Mitigation Methods
6) Identify Risk Recovery Methods
7) Define Risk Metrics
8) Implement Mitigation Actions
9) Track Risk
10) Implement Contingency action

Risk identification team determines Risks which are categorized according to the Risk Levels by mapping each risks onto a Risk Matrix. The risk levels are defined as:

Tolerable Risk
Risk is identified as having little or no effect or consequence on project objectives.

Low Risk
Risk is identified as having minor effects on project objectives.

Identify
Before risks can be managed its must be identified before adversely affecting the project.

Analyse
Analysis is the conversion of risk data into risk decision-making information. It includes reviewing, prioritizing and selecting the most critical risks to address. The Software Risk Evaluation (SRE) Team analyses each identified risk in terms of its consequence on cost, schedule, performance and product quality.

Fig. 1: Risk management Process Paradigm
Medium Risk
Risk is identified as one that could possibly affect project objectives, cost or schedule.

High Risk
Risk is identified as having a high probability of occurrence and the consequences would affect project objectives.

Intolerable Risk
Risk is identified as having a high probability of occurrence and the consequences would have significant impact on cost, schedule and performance.

Risk classification
Risk classification is considered as the proper way to analyse risks and group similar types of risk in different categories or classes. Software risks can be internal risk or external risk.

Risk in software requirements
- No proper requirements gathering.
- No proper documentation.
- The definition of the requirement is very poor.
- On the spot change of requirements.

Risk in software cost
- Poor estimation of the projects cost.
- Poor working of the hardware.
- No proper testing techniques used.
- No proper monitoring.
- Due to the Complexity of architecture.
- Also due to the large size of architecture.

Software scheduling Risk
- Due to the change in requirements and their extension.
- Inadequate knowledge about tools and techniques.
- Due to the lack of manager experience.
Software quality Risk
- Due to the lack of knowledge and the skills.
- Due to the lack of good estimation in projects.

Due to the extension of requirements change.
- Absence of design documentation.
- Due to the inadequate documentation.
- Due to the absence of project standard.
- No proper estimation of budget.
- Poor definition of requirements.
- Due to insufficient knowledge skills.

Review of literature
(Arefeen, 2011) In [1] many software risks have been discussed. The management of the software risk has provided a disciplined environment for decision making to assess continuously the wrong things which are happening and determining the risks which are occurring and the actions which can be taken to deal with these kinds of risks. The objective of managing the risk is to identify problems which had occurred before so that the activities which can help to handle the risk can be planned and invoked as needed. The planning of the risk management addresses the risk management strategies, the processes, methods, the techniques and tools which are to be used to support the process of risk management.

(Basri, 2011) In [2] the author has discussed about the assessment of the risk in software projects development. Risk has been

<table>
<thead>
<tr>
<th>Probability</th>
<th>Frequent</th>
<th>Probable</th>
<th>Occasional</th>
<th>Remote</th>
<th>Improbable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sev.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catastrophic</td>
<td>IN</td>
<td>IN</td>
<td>IN</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>Critical</td>
<td>IN</td>
<td>IN</td>
<td>H</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>Serious</td>
<td>H</td>
<td>H</td>
<td>M</td>
<td>L</td>
<td>I</td>
</tr>
<tr>
<td>Minor</td>
<td>M</td>
<td>M</td>
<td>L</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Negligible</td>
<td>M</td>
<td>L</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

LEGEND
T=Tolerable, L=Low, M=Medium, H=High, IN=Intolerable

<table>
<thead>
<tr>
<th>Probability</th>
<th>Description</th>
<th>Severity</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent</td>
<td>Not surprised, will occur several times (Frequency per year &gt; 1)</td>
<td>Catastrophic</td>
<td>Greater than 6 month slip in schedule; greater than 10% cost overrun; greater than 10% reduction in product functionality</td>
</tr>
<tr>
<td>Probable</td>
<td>Occur repeatedly; an event to be expected (Frequency per year 1-10-1)</td>
<td>Critical</td>
<td>Less than 6 month slip in schedule; less than 10% cost overrun; less than 10% reduction in product functionality</td>
</tr>
<tr>
<td>Occasional</td>
<td>Could occur sometime (Frequency per year 1-10-1)</td>
<td>Serious</td>
<td>Less than 2 month slip in schedule; less than 5% cost overrun; less than 5% reduction in product functionality</td>
</tr>
<tr>
<td>Remote</td>
<td>Unlikely though conceivable (Frequency per year 10-2 - 10-4)</td>
<td>Minor</td>
<td>Less than 1 month slip in schedule; less than 2% cost overrun; less than 2% reduction in product functionality</td>
</tr>
<tr>
<td>Improbable</td>
<td>So unlikely that probability is close to zero (Frequency per year 10-4 - 10-6)</td>
<td>Negligible</td>
<td>Negligible impact on program</td>
</tr>
</tbody>
</table>

Fig. 3: Sample Risk Matrixes
explained as the probability of the loss suffered which shows the impact on the project. This can be stated as a worst quality of the software solutions and results. The essential risk factors affect the software projects which lead to the unsuccessfulness of the software projects. Risk can be reduced and managed in accordance with tight planning and assessment. To formulate different types of risk the risk management is equally divide into two categories i.e. risk controlling and risk assessment. Firstly the risk control can be divided into three sub parts i.e. planning of risk management, risk monitoring and risk resolution. Secondly the assessment of the risk is also dividing into three sub parts i.e. analysing the risk, identifying the risk and prioritizing the risk.

(Shiva Mansoorzadeh, 2011) In [6] the discussion of the author is all about the scheduling of the projects reliability. In this paper the optimization of scheduling will be viewed from the perspective of the reliability. There are many issues related to the scheduling of the project but the two main issues related are a) duration of the project b) uncertainty about the project duration. In the software development process the critical path keeps on changing with the progress of each project. In this reliable project scheduling approach which has taken in consideration the integration of risk management and critical chain schedule analysis both. In the method which is proposed has analysed the risk of the potential projects and the strategies of their responses which are developed by the failure mode of fuzzy and the effective analysis.

(Yogini Bazar, March 2012) In [7] the author in this paper has discussed about the models of the risk assessment at this time software engineering, there is static and dynamic environment of the software development which leads to several faults and errors which are to be handled and taken care of. Managing a risk means a solution that helps to understand the software team and manage the uncertainties. It is important to measure the level of uncertainty and the degree of loss associated with each risk when the risks are analysed. Everyone involved in the software development projects participate in risk analysis and management. Due to poor risk management and poor estimation of budget many software development projects miss their goals of delivering a acceptable software product. The author in this paper has discussed the software models of risk assessment and their comparison according to various risk elements.

(Dr. M. M. Sharma, 2013) In [3] this paper it is briefly discussed about the tools which help to remove risk that are presented in software engineering. Managing risk is the figurative process in which the risk factors are identified systematically, being assessed and are being mitigated. In a project management the toughest part is to find out the risk in the software project. As it is believed that no IT Project can ever be remain free of errors or risk. In this paper, author proposed a framework of automated risk removal model. This model helps to identifies, classifies and generates solution to solve the problems. These problems caused by software risks. This is a phase wise process of risk removal. This is a phase wise process of risk removal. First, risk sources and risks are identified, after that they are classified, and then solution patches are generated and applied on encountered problem.

(Julio Menezes Jr., APRIL 2013) [4] Project management is very necessary while developing a project. For developing a project, the project manager faces the many difficulties related to project failure, risk assessment and so on. Project management is related to one of the discipline of management of risk whose usage is becoming necessary as in the increase of the size and growing in the complexity of software. The software development environments has a great importance for knowing the usage of processes, tools that are being used in the risk management and the techniques used. The resources are shared if the different projects are executed simultaneously one after another. In this paper the indicators which are specific to the software project environment are defined which helps in supporting the risk assessment activities, risk analysis and identification of risk.
CONCLUSION

Now a days, present risk identification tools do not clearly address where the risk is coming from and how the problems caused by the risks and what can be taken to the solution. Available techniques do not provide automated software risk removal.

Automated risk identification tool is novel method which gives different phases of risk identification. In present work lots of methods are used for risk identification. They starts from first phase of identification of software risks and its sources and after then they are identified and analysed before going into next phase. Once the risks are identified then it can be easily classified that it’s as probability of low level risk, probability of medium low level risk, probability of medium level risk, probability of medium high level risk, and probability of high level risk.

Proposed work

The proposed work can be done on the basis of experience of software development experts. We can develop a tool which can help the new software developers to overcome different risks during software development. The proposed tool will help them to tell them that if they choose this way then they may face this type of risk. For example if developer is adding number of very complex algorithms in software then they can create risk for low performance due to high complexity. Now our purposed tool contains all factors that can cause for risk creation and from this tool developer can easily check that what kind of risks he can face during development means this tool will gives the estimation of risk occurrence.

REFERENCES