Strength and Weakness of Software Risk Assessment Tools

Abdullahi Mohamud Sharif\(^1\), Shuib Basri\(^2\) and Hassan Osman Ali\(^3\)

\(^1\)Universiti Teknologi PETRONAS, \(^2\)Universiti Teknologi Malaysia

\(^1\)msharif_g01396@utp.edu.my, \(^2\)shuib_basri@petronas.com.my, \(^3\)xaseey@gmail.com

Abstract

There are different models and methods in the literature that assess risks in software projects. But few of these models and methods propose visualized software risk assessment tools that help project managers to visually identify, analyze, and prioritize risks in software projects. In this paper, a summary of these tools are presented with a detailed explanation of the strength and weaknesses of these software risk assessment tools that are found in the literature.

Keywords: Risk, Software Risk Identification, Software Risk Assessment Tool, Software Risk Analyzing, Software Risk Prioritization, Software Development

1. Introduction

In general, risk is defined as “the possibility of suffering loss that describes the impact on the project which could be in the form of poor quality of software solution, increased costs, failure, or delayed completion” [1]. Moreover, all projects share some degree of risk, and most Information Technology (IT) projects have considerable risks [2]. Risk can, however, be reduced [2], stewarded [3], and managed according to tight planning and assessment.

Software risk assessment is the process of identifying, analyzing, and prioritizing risks that impacts the project. Therefore, in order to prevent or reduce the impact of the risk, a necessary step is to identify the risk factors that can cause fatal effect to the project. As stated in [4], risk identification is the first step in the risk assessment in the domain of risk management domain.

A recent survey carried out by Sharif et al., [5] shows that 71.5% of Malaysian SME companies either do not do any risk assessment for their projects or they only do for certain project. Despite the fact that there are various means of identifying risks including the use of tools, 62.9% of the surveyed companies do not use any tool for their risk assessment [5]. This indicates that there is lack of knowledge of utilizing tools in risk assessment, amongst other possible reasons.

The survey study [5] was carried out also to investigate the reason why the software companies do not use tools for their risk assessment. The result showed that 24% of the software companies do not use risk assessment tools because of cost. Also, 19.4% of the software companies gave experience factor as the reason for not using them. Other related reason factors are: training factor (17.2%), requirement factor (13.1%), staff factor (13.1%), and time factor (12.7%).

In the literature, several studies aimed to reduce the impact of risk is software projects have developed various models and methods using different types of software projects and applications [6–21]. However majority of these models and/or methods are not implemented into visual tools to make their usage easy and friendly. This finding was reported in our
previous studies [22, 23] in which a systematic literature review was carried out to analyze the risk assessment models and methods in the literature.

This paper focuses on analyzing the strength and weaknesses of existing software risk assessment tools in the literature.

The other part of this paper is structured as follows: a review of software risk assessment tools in the literature is presented in Section 2. The assessment procedures followed by these tools are detailed in Section 3. Section 4 presents a detailed discussion of the strengths and weaknesses of the tools. Finally, Section 5 concludes the paper and gives future directions.

2. Software Risk Assessment Tools

According to Hall [24], one of the challenges to accurately manage risk analysis is in utilizing automated tools to store, organize and process data into meaningful information. Yet, very few studies engaged in designing visualization tools for managing risks [25]. We identified only 5 tools implemented for proposed models or methods in the literature. Table 1 describes the identified tools.

<table>
<thead>
<tr>
<th>Author(s) and Year</th>
<th>Area</th>
<th>Model based</th>
<th>Application Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choetkieritkul and Sunetnanta (2012)[21]</td>
<td>Offshore</td>
<td>CMMI Quantitative Approach</td>
<td>IBM Jazz</td>
</tr>
<tr>
<td>Mustafa et al. (2010)[13]</td>
<td>Generic</td>
<td>Probabilistic Inference Model</td>
<td>Unknown</td>
</tr>
<tr>
<td>Sharif and Abd Rozan (2010)[18]</td>
<td>Project Time</td>
<td>None</td>
<td>Oracle Appex</td>
</tr>
<tr>
<td>Iranmanesh et al. (2009) [10]</td>
<td>Generic</td>
<td>Fuzzy logic</td>
<td>MATLAB</td>
</tr>
<tr>
<td>Jianyi et al. (2008)[11]</td>
<td>Generic</td>
<td>Analytic Hierarchy Process (AHP) and Support Vector Regression (SVR)</td>
<td>Delphi 7.0 and LibSVM 2.85</td>
</tr>
</tbody>
</table>

As shown in Table 1, three of the tools are focused on general software projects while the other tools focused on specific type or parts of the software projects. In [20], the authors proposed a model and its tool for assessing risk in remote software development projects or offshoring projects. While in [18], the authors proposed a model that focuses on project time risks which is part of the general project management knowledge areas [26].

Furthermore, these tools were proposed between 2008 and 2012, and they are based on different existing approaches such as Capacity Maturity Model Integration (CMMI) by adapting CMMI best practices [20]. Fuzzy logic [10] and Probabilistic Inference Model [12] were also based on risk assessment models being proposed and visualized as shown in above table. While other tool based their model on Analytic Hierarchy Process (AHP) evaluation model [11].

Moreover, the tools are developed or implemented using different application development platforms. IBM’s Rational Team Concert (RTC) was used by [21] and Oracle Application Express (Apex) 3.2 – a website development and Oracle Database 11g for the database
development – was used by [18]. Other researchers used MATLAB [10], and Delphi 7.0 [11] for their tool development.

Generally, according to Boehm [4], risk assessment are divided into three categories which are: risk identification, risk analyzation, and risk prioritization. In each level, there are different techniques and approaches that researchers can apply in order to identify, analyze, and prioritize risks in software projects. In the following section, the processes in which these tools are adapted in order to determine their strength and weakness in software risk assessment process are discussed.

3. Assessment Procedures in the Tools

In general, software risk assessment is the process of identifying, analyzing, and prioritizing risks in the project. For each process, although there are no convention techniques for these processes, there are different techniques and practices to follow. Therefore, in this section, a brief description of the tool’s processes of assessing risks is being explained and then analyzed them based on the general risk assessment processes with their known techniques available in the literature.

First of all, Choetkiertikul and Sunetnanta [21] developed a tool using CMMI quantitative approach. The tool uses the data and evidence of the project management stored in Jazz repository to initiate assessment of risks in the project. The tool follows these steps for identifying, analyzing, and prioritizing the risks in the project:

- Risks are stored in the repository before assessment is being initiated
- There is Extraction Engine which collects required project management evidence from the repository in order to analyze them.
- An Assessment Engine will calculate the degree of the risk for the collected evidences.
- The Engine also calculates the Quality of Project (QoP) and compares to related CMMI practice of each risk factor and displays in the dashboard. The more QoP a project has, the less project risk would be.
- The tool displays two graphs which the first one shows historical view of the assessed risks remotely towards the QoP. The second graph shows potential and critical risks in the project.

On the other hand, the tool proposed by Mustafa et al., [13] is called Risk Assessment Visualization Tool (RAVT). The risks in the tool are identified from experienced project managers through questionnaire or past projects data. The tool uses nine critical risk elements to categories risks in the questionnaire and the tool assesses the risks by following these steps:

- A list of questions for each risk category is distributed to experience project managers.
- Each question contains three choices by each choice carries 1, 2, or 3 mark respectively.
- After answering the questionnaire, the tool calculates and displays the risk by adding all marks for each category and normalizing it by dividing the total questions for each category.

For another tool, Sharif and Abd Rozan [18] proposed a tool called Risk Assessment Tool (RAT) which assesses software project time risks. The tool automatically assesses the risk of the project based on the project documentation and project tasks. Following, are the steps that RAT follows in order to identify, analyze, and prioritize risks in the project:
The tool allows creating projects and filling the information about the start, finish dates of the project, Lines of Code (LOC) of the project, budget, and number of project staff.

After initiating the project, the project manager adds tasks to the project while each task has its own information like task start, finish dates, task resources, and task owner.

The Risk Fetching process calculates the risk of the project based on the project plan uploaded at previous steps. This process calls Risk Rules that combines conditions, scenarios, and ranks of each risk.

The tool displays a table format of all risks that being identified from the project plan by ranking them based on Risk Rank Matrix which is being calculated from the category, probability, and impact of the risk.

Furthermore, the proposed tool Iranmanesh et al., [10] is based on fuzzy expert system and being developed by using MATLAB. The tool evaluates the risk of software projects by all of its aspects by using fuzzy rule base approach. The tool follows these steps to identify, analyze, and prioritize risks in software projects:

- Adapted risk categories were used to be analyzed in the tool.
- The tool determines fuzzy sets and membership functions for each risk category.
- A Rule Base is used which consists of two layers; the first layer contains risk matrix and the second layer contains the collected risk factors.
- The tool outputs numerical values which present state of risk for each category as well as the risk of project called the total risk.

On the other hand, the proposed tool by Jianyi et al., [11] is called Project Risk Assessment Decision Support System (PRADSS). The tool consists of risk index system, AHP evaluation model, intelligent evaluation system, knowledge base and man-machine interaction. The tool integrates Analytic Hierarchy Process (AHP), Support Vector Regression (SVR), and Decision Support System (DSS). The proposed tool follows below steps to assess risk for software projects:

- The risks are identified outside of the system by using different techniques of risk identification and then inserted to the tool by following specific patterns.
- Risk Index System is initiated which calculated the weight for each risk factor that being collected by using AHP algorithm.
- The bottom level of the AHP hierarchy is evaluated individually and assigned points according to each unique criterion. The overall project risk is weigh by taking consideration of probability and impact risk.
- The AHP Evaluation model selects suitable risk factors and forecasts risks of the project with their impact.
- Then the risks are presented in a table format by ranking them using scope of risk index.

Finally, for the above briefed tool’s procedures in assessing risks, it paves the way by providing explanations on the strengths and weakness that these tools have, based on the software risk identification, analysis, and prioritization processes they followed. A detailed of these strengths and weakness are discusses in next section.
4. Strength and Weakness of the Tools

In this section, researchers examine the strength and weakness of the above mentioned tools based on software risk assessment processes. As mentioned earlier, currently there are different tools with models and methods proposed in the literature and can be classified into certain number of categories in order to assess and identify their weakness and strengths. Table 2 summarizes the strength and weakness of the recent software risk assessment tools in the literature.

Table 2. Strength and Weakness of Current Software Risk Assessment Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
</table>
| [21] | • Combination provides better approach for analyzing the risk  
   • Provide suitable guidelines to each risk  
   • Risk can be assessed on different phases of the project | • All risks cannot be identified automatically like risks related to the organizational environment  
   • Focuses only the quality of the project  
   • User’s involvement of risk identification is not considered  
   • Does not have a way to add, update, or delete risks  
   • No risk prioritization |
| [13] | • Adding, updating, or deleting any risk questions or category  
   • Uses probability inference model as a way to represent the subjective judgments of experienced software project managers | • Risk identification is based only on questionnaire or list of questions  
   • User’s involvement of risk identification is not considered  
   • Risk analysis is poor  
   • No prioritization of risks |
| [18] | • Project user’s involvement of risk identification is considered  
   • Analyzing of project time risk are based on risk rules  
   • Adding, updating, or deleting any risk is possible  
   • Usage of ranking matrix | • Only based on project time risks  
   • All risks cannot be identified automatically like risks related to the organizational environment  
   • There is no risk factors allowed  
   • Risk dimensions or categorization of the risk is not considered |
| [10] | • Allows comparison of project risks for different projects  
   • Uses fuzzy rules to analyze the risk | • Not identifying any risk factor  
   • Adding, updating, or deleting is not available  
   • Not prioritizing the risk categories |
| [11] | • Allows comparison of project risks for different projects  
   • Integrates AHP, SVR, and DSS  
   • Prioritizes the risks | • Risk identification techniques used were not part of the tool  
   • Risks cannot manually added, updated, or deleted  
   • No risk dimensions or categorization |

First of all, the tool developed by [21] is based on quantitative CMMI risk assessment model which groups the risk based on risk taxonomy and then maps offshoring risks into specific CMMI practice.
• Although the users are not participating in risk identification process but the tool uses different users for different aspects. The users of the tool are Project, Managers, Assessment Experts, and Project Team.

• The tool analyses the risk by collecting risks from the project data and grouping them using risk taxonomy-based approach with attaching each risk to a suitable CMMI best practice. This combination provides better approach for analyzing the risk by providing suitable guidelines to each risk.

• Risk can be assessed based on different phases of the project not only the initial phase of the project.

However, this tool is based on offshore projects or projects that development takes remotely in different locations and requires risk assessment. Remote development can take place in offices in different countries or in the same country. This type of development is increasing as companies spread in different regions by operating different branches in different states or countries. When compared the software risk assessment process to the developed tool by [21] we figure out:

• The tool identifies risks automatically based on project data stored in the repository. The tool uses data like work breakdown structure, requirement reviews, test documents, etc. In software development, there are risks that cannot be identified automatically like risks related to the organizational environment that related to the project. Therefore, this tool only focuses risks that can be identified systematically from the documents of the project.

• The tool focuses only the quality of the project

• The project user’s involvement of risk identification is not considered

• When new risk arises the tool does not have a way to add, update, or delete risks

• The tool only displays two graphs which contain the total number of risks in graphical way without prioritizing them

On the other hand, The Risk Assessment Visualization Tool (RAVT) that is being developed by [13] is generic tool which can be applied to any software risk assessment.

• The tool uses probability inference model as a way to represent the subjective judgments of experienced software project managers

• Although the user involvement in risk identification is missing, the tool involves experienced software project managers in risk identification process

• The system provides possibility of adding, updating, or deleting any risk questions or category

As the authors mentioned, the RAVT tool is not yet completed [13]. Simulating the normalized risk value for each category into graphical way and displaying the overall project risk level are their future work. But based on the software risk assessment process, the RAVT tool although it’s generic and applies to all types of projects but a necessary steps are missing:

• The tool uses risk categories which contains sub questions in three choices per each question. That means risk identification is based on questionnaire or list of questions

• The project user’s involvement of risk identification is not considered
• Risk analyzing is based on marking choices for each question from 1, 2, and 3 which does not differentiate the importance of each sub question

• There is no prioritization of risks

Moreover, the Risk Assessment Tool (RAT) which is being developed by [18] is a generic tool for project time risk assessment tool. The tool focuses on the management aspect of the project risks by systematically identifying, analyzing, and prioritizing time risk of the project based on project documentation and project tasks like work break down structure.

• The project user’s involvement of risk identification is considered in this tool along the project managers and stakeholders

• The analyzing of project time risk are based on risk rules which are combination of scenarios, conditions, and rank for each risk

• The system provides possibility of adding, updating, or deleting any risk after identifying project time risks

• The tool uses ranking matrix to rank the analyzed risk based on category of the risk (1-Unknown, 2-Low, 3-Medium, 4-High, 5-Fatal), probability of risk, and the risk impact to the project (1-Low, 2-Medium, 3-High)

However, RAT system identifies risks automatically without the intervention of human factor. Which creates an important gap as involving experts in risk identification is important. Although the tool allows the project managers to delete or add manually risks to the final list of fetched project risks but there are some weaknesses to the tool which are:

• The tool is only based on project time risks while there are other risks in software project which have their own impact to the project.

• Risk identification is automatic and there is no risk factors allowed the project team to identify or use existing risk factors from the other projects.

• In software development, there are risks that cannot be identified automatically like risks related to the organizational environment that related to the project. Therefore, this tool only focuses risks that can be identified systematically from the documents of the project.

• Risk dimensions or categorization of the risk is not considered in this tool

Additionally, the expert system that developed by [10] calculates the project risk by receiving probability of failure and severity of impact as numerical values and after two layer fuzzy inference returns the risk of each risk category as well as the total risk of the project.

• The tool allows comparison of project risks for different projects

• The tool uses 17 million fuzzy rules to analyze the risk of the 12 risk category that being adapted as well as the total risk of the project

Nevertheless, this expert system is being coded by using MATLAB fuzzy sets. Considering the software risk assessment processes, we can summarize from this tool:

• The tool is not identifying any risk factor

• The tool only analysis the total risk of the project by only analyzing the risk dimensions or categories that adapted from [27] without identifying or adapting the risks in these categories
• Adding, updating, or deleting a risk factor option is not available
• Although there is no risks to prioritize but also the system is not prioritizing the risk categories

Researchers in [11] proposed a tool called Project Risk Assessment Decision Support System (PRADSS) and it is being implemented by using Delphi 7.0 and LibSVM 2.85.
• The tool allows comparison of project risks for different projects
• The tool integrates Analytic Hierarchy Process (AHP), Support Vector Regression (SVR), and Decision Support System (DSS) to deal with quantitative project risk assessment problem under knowledge based intelligent architecture
• The tool prioritizes the risks of the projects

But this tool is applicable to software projects that are based on decision support systems. Although the tool is generic and can be applied to different types of software projects but the researchers considered some important points about the tool regarding the software risk assessment processes and they are:
• Risk identification techniques used were not part of the tool
• The risks are inserted through specific pattern and there are no options that risks can manually being added, updated, or deleted if required.
• Risk dimensions or categorization of the risk is not considered in this tool

Finally, as in the above table and followed explanation, researchers elaborated in detail the strength and weakness of current software risk assessment tools in the literature.

5. Conclusion and Future Work

In this paper, researchers assessed the recent software risk assessment tools in the literature by examining their strength and weakness based on the software risk assessment processes. There are some weaknesses that were mostly common for all tools such as: risk prioritizing process was not either available or not adequate, and risk identification techniques were not applied.

These tools are based on different models or methods and in order to know further about the strengths and weakness of these tools, researchers are currently studying the models or methods of these tools to examine the input, output, activities, and roles of these models and methods used and their assessing issues while at the end proposing an integrated model that fills the identified gaps in the literature.

References


Authors

Abdullahi Mohamud Sharif, received his B.sc. in Computer Science from Mogadishu University (MU), Somalia in 2007 and M.sc. in IT Management from University Technology Malaysia (UTM), Malaysia in 2009. He has experience as lecturer and as Analyst Programmer from different organization. Also he is PRINCE2 Practitioner. Currently he is doing his PhD in Information Technology from Universiti Teknologi PETRONAS (UTP), Malaysia.
Shuib Basri, currently is a Senior Lecture at Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia. He received his PhD from Dublin City University, Ireland in 2010. He earned his Master degree from University of Melbourne, Australia. His research interests include software engineering, software process improvement, software quality, and knowledge management.

Hassan Osman Ali, currently, he is doing his PhD of Information Systems at Universiti Teknologi Malaysia (UTM). He receive his M.sc. of IT Management from Universiti Teknologi Malaysia (UTM) on 2010. He also graduated his degree of Computer Science and Information Technology from Mogadishu University (MU), Somali on 2007. His employment included the Nationlink Telecommunication Company in department of GSM system on 2007. Additionally, he worked with Qaran Money Transfer Company on 2008. Now, He is Certified Prince2 Project Management on 2012.