GLOBAL ACADEMIC RESEARCH INSTITUTE

COLOMBO, SRI LANKA

International Journal of Technological Advancement and Research

ISSN 2424-6395

Volume: 01 | Issue: 01

On 30th June 2022

http://www.research.lk

Author: Dr. Rasika Aponsu

IIC University of Technology, Cambodia

GARI Publisher | Software Engineering | Volume: 01 | Issue: 01

Article ID: IN/GARI/JOU/2022/119 | Pages: 25-31 (06)

ISSN 2424-6395 | Edit: GARI Editorial Team

Received: 24.04.2022 | Published: 30.06.2022

THE UTILISATION OF THE RISK SENSITIVITY ANALYSIS APPROACH FOR VISUALISING SOFTWARE RISKS IN SOFTWARE ENGINEERING PROJECTS

Dr. Rasika Aponsu

IIC University of Technology

Cambodia

ABSTRACT

The practise of project risk management entails a methodical approach to the identification, analysis, and mitigation of software-related risks within a project. Regrettably, software engineering initiatives were plagued with a multitude of software hazards. Consequently, a significant amount of time was required to thoroughly evaluate the various software risks that may arise within the project. The majority of project managers demonstrated limited awareness of software hazards until such risks manifested themselves during the course of the project. Hence, it is imperative to create an overview that employs sensitivity analysis to visualise and prioritise software risks. This will aid project managers in identifying software risks that have manifested in software engineering projects. This research employs the Systematic Literature Review (SLR) technique to identify the software risks that have happened in the project. Additionally, a web-based integrated approach with risk sensitivity analysis is utilised to visualise and prioritise the software risks. The research proposes that the utilisation of a web-based application, together with risk sensitivity analysis, can serve as a viable approach to aid project managers in the identification of software risks with significant potential in software engineering projects.

Key Words: Software risks, Sensitivity Analysis, Visualisation, Risk Sensitivity Analysis, Risk management

INTRODUCTION

The implementation of project risk management is of utmost importance in the context of software projects that are being handled. The success or failure of a project can be determined (Fauzi & Sanim, 2014). The management of project risks encompasses a methodical procedure that entails the identification, analysis, and response to project hazards (Marchewka, 2003). The analysis of past software risks is conducted, followed by the implementation of targeted measures to mitigate the likelihood of future occurrences (Fauzi, Ramli, & Nasir, 2008). Software risks refer to unpredictable events that have the potential to adversely impact project objectives, should they materialise (Marchewka, 2003). Regrettably, project managers consistently commit numerous typical errors when managing software risk within projects. Consequently, the project managers experience delays in identifying software risks that possess a significant likelihood of occurrence and are of utmost importance. The mitigation of these errors can be achieved through the application of the risk sensitivity analysis approach, which involves visualising the software hazards. Risk sensitivity analysis is a methodology employed to discern the significant uncertainties with the aim of visualising the process of data collecting (Cullen & Frey, 1999). Risk sensitivity analysis is not a methodology aimed at

completely mitigating risks under any circumstances. This technique serves the purpose of establishing a theoretical framework and offering informed visualisation of project risks to project managers.

This study presents a web-based tool that incorporates a risk sensitivity analysis approach. The utilisation of web-based applications can aid project managers in visualising and prioritising software risks that may arise in software engineering projects. The initial half of this paper will provide an introduction to the concept of software risks within the context of software engineering projects. Subsequently, the paper proceeds to elaborate on the employed approach. The findings of this investigation are reported in the Results section. Subsequently, the outcome is deliberated upon. The final section of the study serves as the conclusion, aptly titled the Conclusion Section.

Research Background

The software engineering project encompasses several software risks, necessitating a substantial investment of effort to thoroughly examine all potential risks that may arise throughout the duration of the project. In the realm of software risk assessment, two primary categories of hazards can be identified: internal risks and external risks. Internal risks refer to software hazards that originate within an organisation and possess the potential to jeopardise a project. On the contrary, external risks refer to software risks that originate from sources external to the organisation, posing challenges in terms of control (Asif, Ahmed, & Hannan, 2014). Internal risks can be categorised into three main types: management risks, people risks, and financial risks. Technology risks, requirement risks, and estimating risks are categorised as external risks.

Risk management is a critical aspect that holds significant importance for project managers. The assessment of management risks serves as a means to evaluate the project manager's leadership abilities. According to Makhani et al. (2010), it may be argued that a proficient project manager possesses the skills and competencies necessary to effectively manage risks associated with a project. According to Bannerman (2008) and Wallace, Keil, and Rai (2004), the project manager's ability to effectively manage software risks and develop appropriate risk management strategies helps mitigate the impact of software risks on project progress and overall project success. According to Kremljak and Kafol (2014), some instances of management risks include inadequate internal communication, ambiguous project scope, and the presence of an inexperienced project manager.

The consideration of people risk is a fundamental aspect in the process of project creation and decision-making. Individuals that participate in project development are anticipated to possess qualities of being cost-effective and compassionate. However, their inherent characteristics also enable them to satisfy many motivations in making business decisions (Kremljak & Kafol, 2014). Some examples of hazards related to individuals in a project include team members that lack the essential skill-set, the loss of critical staff at a crucial moment in the project, and team members who are unmotivated. Financial risks are associated with the potential occurrence of events that have a detrimental impact on the overall financial feasibility of a project (Kremljak & Kafol, 2014). Examples of financial risks include the expenses associated with the allocation of resources for the development of a growing project, exceeding budgetary limits, and an escalation in staff remuneration. Technology hazards refer to the potential

for experiencing negative outcomes as a result of engaging in technological activities. Technology risks refer to the various hazards that arise as a result of software and hardware failures, software security vulnerabilities, and the inappropriate utilisation of software and hardware technologies throughout the creation of a system (Younis, Malaiya & Ray, 2016). There are other instances of technological hazards that can be identified, including the inadequacy of technology and tools, the loss of vital data due to software errors, and the insufficiency of processor memory.

Requirement risks refer to adverse occurrences that are associated with unclear or ambiguous requirements, as well as changes in user needs that occur throughout the project development process. Requirement hazards typically pertain to the direct association with the project's specifications. In addition to the aforementioned considerations, it is important to acknowledge that requirement risks possess the potential to exert adverse effects on the financial aspects, timely completion, and overall quality of a project, should they materialise. The potential hazards associated with requirements can include inadequate stakeholder engagement, new requirements introduced by the client, and changes in project requirements (Arnaut, Ferrari, & Souza, 2016). Estimation hazards refer to the process of quantitatively describing the length, intensity, quantity, and potential effects associated with a software risk (Lewis, 2001). Examples of estimating hazards include setting unrealistic timelines, allocating an inappropriate budget, and mismanaging resources.

Hence, it is imperative for the project manager to possess a comprehensive understanding of the potential software risks that may arise during the course of the project. The purpose of this is to ensure effective management of potential software risks, hence mitigating any negative impact on the project. Evidence indicated that the project manager failed to implement a systematic approach to software risk management in their project, despite their assertion of having done so.

According to Bannerman (2008), a literature review indicates that the majority of project managers possess knowledge of software hazards only after these risks have manifested themselves during the course of their projects. The aforementioned issues have emerged due to the limited availability of methods that enable the visual identification and analysis of software risks at the early stages of a project (Sharif, Basri, & Ali, 2014). Risk sensitivity analysis is a method utilised by project managers to discern the significance of various software hazards, distinguishing between those that have importance and those that do not (Smith et al., 2008). Consequently, by the use of risk sensitivity analysis, the project manager can obtain an earlier forecast on the project's vulnerability to software risks (Makhani et al., 2010). However, there is a limited amount of research that utilises the risk sensitivity analysis approach for the purpose of managing software risks in software engineering projects. This study presents a web-based tool that aims to offer project managers an overview of software risks that could potentially impact the progress of software engineering projects. The proposed approach utilises risk sensitivity analysis to assess these risks.

METHODOLOGY

The study is divided into two distinct phases: the Systematic Literature Review (SLR) and the creation of the application.

The initial step of the study involved the utilisation of Systematic Literature Review (SLR) methodology to synthesise and identify software hazards that manifest in software engineering projects.

During the second phase, a web-based application is developed that incorporates a risk sensitivity analysis approach. The equation presented below is commonly employed in software engineering projects to quantitatively assess and prioritise software risks:

RE = Probability of occurrences (UE) * Probability of consequences (UE) Where, UE = Uncertainty event

RESULTS

Table 2 is a comprehensive depiction of the software hazards that were encountered in software engineering initiatives.

The visual representation of software risks through sensitivity analysis is depicted in Figure 1 and Figure 2. These figures showcase a web-based application designed specifically for visualising and prioritising software risks encountered in software engineering projects.

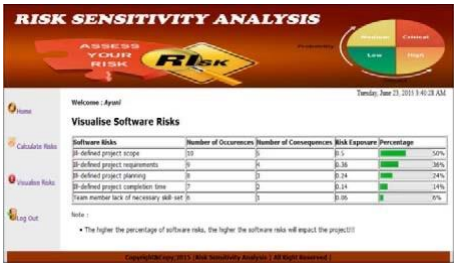

Figure 1. Visualise Software Risks in Software Engineering Projects

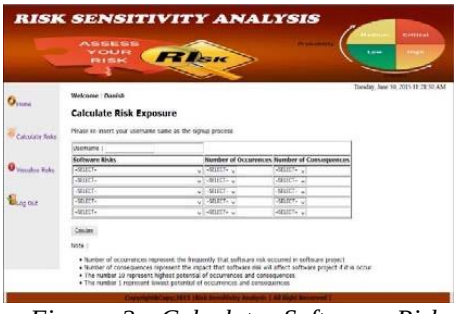

Figure 2. Calculate Software Risks Page

DISCUSSION

The findings derived from Table 1 indicate that the identified software risks manifest themselves in software engineering initiatives. The findings indicate that the occurrence of management risks is higher in comparison to both people risks and technology risks. The occurrence rate of management risks is 66%, whilst people risks account for 25% and technological risks constitute a mere 12%. The significance of management risks in software engineering projects cannot be overstated. The occurrence of management hazards might result from errors made in the management of a software engineering project. The findings presented in Table 1 demonstrate that the occurrence of people risks and technology risks is comparatively lower when compared to management risks. The presence of both human and technological hazards necessitates the implementation of a mitigation strategy, which may involve sending team members for training and fostering a cohesive collaboration environment. The findings of this study are of utmost importance in enabling the project manager to gain insight into the potential software hazards that have arisen in software engineering projects.

Figure 1 presents a visual representation of the software risks that have been discovered, with emphasis placed on their significance to the software engineering project. The software risks that have been found are presented in a risk impact table through the utilisation of the risk sensitivity analysis technique. In this manner, the project manager is able to assess the likelihood and potential impact of each software risk. The assessment of the likelihood and potential outcomes of software risks will ascertain the magnitude of risk exposure and the extent of damage in the event that these risks materialise inside the software project. The impact of

software risk on a software project is directly proportional to the value of risk exposure and the percentage of software risk. The utilisation of software risk visualisations serves the purpose of offering project managers a comprehensive understanding of the relative effect levels associated with different software risks inside a software engineering project. Consequently, the project manager may adequately prepare and develop appropriate risk management techniques to mitigate the impact of software risks on the project's advancement.

CONCLUSION

The present study investigates the software hazards that manifest in software engineering initiatives. A compilation of ten scholarly research papers was utilised to synthesise the software dangers that have manifested in software engineering projects. The software risks that were identified have been categorised into three distinct types: management risks, people risks, and technological risks. This study has also identified a total of 39 software dangers that have been observed in various software engineering initiatives. A total of 21 software risks were attributed to management risks, while an additional 9 software risks were associated with people risks. The other 9 software risks were identified as stemming from technology risks. Based on the findings, it can be inferred that management risks exhibit a greater frequency in software engineering projects when compared to both people risks and technology risks. This study holds significance since it employs risk sensitivity analysis to emphasise and visually represent the various software hazards that may arise in software engineering projects. In addition, this study serves to enhance the awareness of project managers regarding potential software risks in software engineering projects.

Additionally, this study has presented a risk impact table that offers project managers a comprehensive understanding of the various software hazards associated with software engineering projects. Therefore, it is essential for the project manager to allocate sufficient time for the formulation of risk management methods in order to mitigate the potential software risks and minimise their impact on the execution of the project. In subsequent research endeavours, it would be advantageous to enhance this study by using a diverse range of methodologies for visualising software hazards. The incorporation of various visual aids such as pie charts, line graphs, and SWOT analysis graphs can enhance the visualisation process, as opposed to solely relying on risk impact tables and horizontal bar graphs. There are several methods of visualisation that can enhance efficacy and provide project managers with a more comprehensive understanding of software risks in software engineering projects.

REFERENCES

Arnaut, B. M., Ferrari, D. B., & e Souza, M. L. D. O. (2016). A requirements engineering and management process in concept phase of complex systems. IEEE International Symposium on Systems Engineering (ISSE). IEEE.

Asif, M., Ahmed, J., & Hannan, A. (2014). Software Risk Factors: A Survey and Software Risk Mitigation Intelligent Decision Network Using Rule Based Technique. International MultiConference of Engineers and Computer Scientists (IMECS).

Bannerman, P. L. (2008). Risk and risk management in software projects: A reassessment. Journal of Systems and Software, 81(12), 2118-2133.

Cullen, A. C., & Frey, H. C. (1999). Probabilistic techniques in exposure

assessment: a handbook for dealing with variability and uncertainty in models and inputs: Springer.

Fauzi, M., & Sanim, S. (2014). Mapping of software configuration management issues with the area of responsibility. IEEE 8th International Conference on Application of Information and Communication Technologies (AICT).

Fauzi, S. S. M., et al. (2010). Software Configuration Management in Global Software Development: A Systematic Map. 17th Asia Pacific Software Engineering Conference (APSEC). IEEE .

Fauzi, S. S. M., Ramli, N., & Nasir, M. (2008). Assessing Software Risk Management practices in a small scale project. International Symposium on Information Technology (ITSIM)..

Kremljak, Z., & Kafol, C. (2014). Types of Risk in a System Engineering Environment and Software Tools for Risk Analysis. Procedia Engineering, 69(0), 177-183.

Lewis, J. P. (2001). Limits to software estimation.ACM SIGSOFT Software Engineering Notes, 26(4), 54-59.

Makhani, S., Khan, A., & Soomro, S. (2010). Project management risk sensitivity analysis. Journal of Information & Communication Technology, 4(Spring), 38-48.

Marchewka, J. T. (2003). Information Technology Project Management–Providing Measurable Organisational Value, 2003: John Wiley and Sons Inc., USA.

Sharif, A. M., Basri, S., & Ali, H. O. (2014). Strength and Weakness of Software Risk Assessment Tools. International Journal of Software Engineering & Its Applications, 8(3).

Smith, E. D., Szidarovszky, F., Karnavas, W. J., & Bahill, A. (2008). Sensitivity analysis, a powerful system validation technique. Open Cybernetics & Systemics Journal, 2, 39-56.

Wallace, L., Keil, M., & Rai, A. (2004). Understanding software project risk: a cluster analysis. Information & Management, 42(1), 115-125.