 doi:10.20944/preprints201912.0128.v1

Comparative Analysis of Risk Management Techniques for Large-Scale Systems

Mehreen Sirshar
Faculty of Software Engineering
Fatima Jinnah Women University
Rawalpindi, Pakistan
mehreensirshar@fjwu.edu.pk

Aliya Shahid
Department of Software Engineering
Fatima Jinnah Women University
Rawalpindi, Pakistan
aliyashahid97@gmail.com

Zoya Alam
Department of Software Engineering
Fatima Jinnah Women University
Rawalpindi, Pakistan
zoyabibi555@gmail.com

***Abstract*—** Risk assessment management have been a hot topic for the researchers since a very long time. Software risk management is an important part of project management as it contains the identification, analysis, estimation and monitoring of different risks present in the system. This helps developers in decision making while assessing the problems that could arise in the software systems. Risk management is very complex in large scale system as these systems have very complex development. The paper describes risk management techniques for large scale system. Furthermore we have provided a detailed comparative analysis of these techniques with commonly identified risks in software systems and have provided a systematic order for risk management process to ensure risk mitigation.

***Keywords*—**large scale systems; risk assessment; risk management techniques; risk mitigation

I. Introduction

Risk is considered as an unsafe incident or situation when it occur it causes the positive or negative effect on the project objective. In creating the high quality software different risk can occur, which cannot be removed entirely but their influence on product can be removed by project manager. In today's world complexity increases in the software system so number of risk occur in a project also increase. Due to high quality risk management is considered more important in future for the large scale system. The main aim of risk management is to detect and control the risk as they occur in project.

There are two types of risk i.e. systematic and un-systematic risk. The risk that are caused by the external factor, viruses, power loss etc. are come under the systematic risk. Vulnerable browser is an example of systematic risk in which loophole may cause security breaches and that harm the organization resources. Generic risk is another name of systematic risk. The second type is un-systematic risk that causes the distinctive risk such as application fault, data defeat, human communications and misuse of important data. Specific risk is another name of un-systematic risk.

Risk management framework consists of risk identification, risk analysis, risk planning, risk tracking, risk control and risk monitoring. First of all risk is detected in any project. For risk identification there are many tools such a brain storming, check list, risk documentation etc. Next phase is risk analysis, in which risk are classified and prioritized. Then next is risk planning, in which different tool are used i.e. human reliability analysis, technical tool analysis etc. Then risk are tracked through the periodic risk mitigation reports and plans. Then risk are controlled by the prototyping, benchmarking, simulation etc. At last risks are monitored through training program and team building session. Also risk are eliminated and reduced by different risk management techniques and tools.

In this paper we have discussed the different approaches that are used to reduce the risk of large scale systems such as "Software Risk Evaluation(SRE) Technique", "Riskit model", "Team Risk Management(TRM) technique", "Analyzer for Reducing Module Operational Risk (ARMOR)", "Softrisk Model" and "CMM based Software Risk Control Optimization Model". Furthermore we have also included a comparative study of different techniques for the large scale system.

II. Literature Review

Software risk management comprises the studies of risk identification, estimation, mitigation and controlling. The paper [19] contains a detailed analysis of software risk factors and the techniques used to eliminate them. Risk measurement is very difficult in large-scale system as compared to small-scale system. So the large system is divided into sub-system and the risk mitigation is possible. This paper defines the variance between large software and small software. Author also discusses the comparative study of different software related risk management models based on common features and categorize them. Risk management process is very important for the large scale software. Following are the steps of risk management process: Prediction: Risk management tools are used for the prediction of the risk, Planning: In this step risk manager compute, evaluate and priories the risk, Risk Analysis: To review the risk, it will possess evaluating the whole of these processes, Risk Monitoring and Reporting: This process will keep the record of risk that are identified and their measuring process, Communication and Feedback: All the stakeholders share their feedback to project managers and it will help project managers to avoid risk and develop a risk free product and Risk Documentation: All of the process of the risk management will be written in a document that will help association. [6]

There is a wide range of challenges and activities for the product owner in large scale agile projects. So in this paper, author describes the activities and behavior to the product owner and line managers. Mainly the product activities and responsibilities are elicitation, prioritizing and approving software. For large scale product the author acknowledged three further sets of activities for product owner. (1) Scale: Product owner is responsible for the

corporation of larger groups and manage association between broader kinds of stakeholders in large scale software. Three steps involved in this process. First step is that project sponsor creates a vision. It responsibility is to keep focusing on project vision. Second step is that, there are intermediaries that have complete knowledge of project goal attained by project sponsor. Third step is to release plan master. (2) Distance: As there are many stakeholders involved in the project, so it is difficult to communicate with them. Each of stakeholders has different temporal, physical and cultural distance. It is the responsibility of product owner to manage them. The communicator is responsible to connect the different stakeholders and resources using different technologies. Maintain a face to face communication between them. (3) Governance: For quality self-governing team share a set of standard. It is the responsibility of governance to maintain the standards. [1]

Artificial Intelligence techniques and algorithms have been used for risk analysis as they can be used for the prediction of any uncertain, vague and incomplete specification. For a project to be successful, identifying the risks involved in its initial state is very critical. Although there are many models available for risk assessment but classification of a risky project is very uncommon. [20]

As we are aware that complex and large scale software systems are more prone to risks so accurate predictive approaches are required to ensure that all the risks involved are identified. Weng Ming Han provides an approach that can be used during early planning of a project for an organization with limited resources. This can help in identifying the underlying risks involved and can prevent significant schedule slippage and expense. The proposed approach uses three layered NN architecture with back propagation algorithm. This models can work on complex patterns and provide results after going through four accuracy evaluation processes and two performance evaluation processes. The model is very accurate as seen from the experimental results and can predict whether the project is risky or not as compared with the previous approach that is using Logical Regression. [3] One of the very first model introduced for the risk assessment was using Logistic Regression Analysis. It was not as accurate but still could be used for the detection of risky projects. [8] A more recent study for software test management proposes a logistic regression based methodology to assess test quality. In this paper, they have introduced an approach to develop metrics outline for test management, and number the description, kind and variety of each and every metric. [9]

Another main problem with risk assessment is risk prioritization. Effective prioritization will determine the overall success of the project and also the life cycle of the project. A study is proposed for the hybridization of Fuzzy Logic multi-criterion decision making approaches to develop a framework that can identify and prioritize the software risks. The method follows a rating mechanism to identify the core risks. The results of this study shows that this is a very effective method for making decisions during the project validation. Comparing the previous approaches used, this approach provides an accurate measurement of significant risk factors in a project. [7]

Shan Liu and Ling Wang have discussed the effects of important strategies and risk on the systems performance. According to the authors the project management risks arise differently in the interior and outsourcing projects. Furthermore the strategic planning effects both types of projects differently. Social subsystem risks have a great effect on implementation in outsourced projects whereas technical subsystem risks have a great effect on implementation in internal projects. The effect of project management risk is equal in both kinds of projects. [5] Another paper provides a study of the effect of risks on the formal and informal control on the process performance of IT projects. It clearly demonstrates that risk moderates the formal and informal control on the process performance of IT projects. [4]

Another paper discusses the recent testing standards and the techniques that are used for the risk based testing. Risk-based testing helps developers to identify and test modules of a system for detecting crucial faults early rather than detecting them after deployment. The present test standards defined have increasingly suggested risk-based testing. In this paper, the authors have thoroughly analyzed the requirements to combine testing and risk assessment coming from three test standards. These test standards are from related bodies like “ISO”, “ETSI” and “OWASP”. They systematically describe which standards recommend using risk-based testing techniques and procedures in specific or specific test areas using a risk-based test taxonomy. In addition, established approaches and approaches derived from original research show how to meet the requirements of the standard and go beyond these recommendations. [2]

The risks in software projects can also be related to finance. A paper has discussed these risks and proposed a solution using Reference Class Forecasting Technique. The work demonstrated can be used by different organizations during risk assessment process in initial stage. This can help them estimate a correct budget for their project development. [10] Another paper discuss the risk assessment techniques in system-of-system (SoS). As these systems are very complex so a framework is proposed that eliminated the major risks that arise due to the complexity and interdependencies of the system. [11]

III. Risk Mangemnet Techniques

A. Software Risk Evaluation (SRE) Technique

Software risk evaluation is the practice of strategic identification and development to reduce risks during the development of a software system. It is basically a decision making tool for the system being developed. This techniques has a lot of risk managing activities that reduce the occurrence of risks after the system has been developed. This is really beneficial for the system as when integrated with other existing tools and techniques, it can enhance the current project management practices. Using this technique the risks are first identified in all the stages of the development cycle of the software system i.e. product, process and the constraints applied. Then these risks are categorized into levels. The risks in each level are

prioritized in the order of most critical to the system to the least critical to the system.

Software risk evaluation helps the developers to identify such risks that are critical to the project initialization. It covers all the aspects of risk management including risk identification, categorization, prioritization, analysis and mitigation planning. This allows the project developing team to have a clear and understandable view on the risk that can be subjected to the system development.

B. *Riskit Model*

This tool uses graphical representation of risks. It can prioritize risks based on previous data that it has been trained on. It also performs accuracy assessment and supports multiple stakeholder and goals. This is a formal method as it generate an analysis graph in which all the risks are explicitly identified and defined. This method has been used in multiple projects in Europe and America. This is a very practical and systematic risk management approach which provides qualitative analysis of risks identified.

C. *Team Risk Management (TRM) Technique*

According to the a research this technique basically provides a number of techniques, processes and support that allows both the organization that the system is being developed for and the organization that is developing the system to actively participate in the decision making processes. It defines the legislative structure and operative events for risk management during the development of the software system, in its entire sections of life-cycle. This ensures maximum participation from all the individuals involved in the development of the software system.

This technique has only four processes that covers the SEI standards processes. The entire process includes identifying risks, analyzing and reviewing them regularly, plan risk mitigation, tracking the risks and communicating the risk information among all individuals involved. This is a continuous process as the risks do not exists at only one stage of project development. This techniques performs routine risk identification and analysis.

D. *Analyzer for Reducing Module Operational Risk (ARMOR)*

Analyzer for Reducing Module Operational Risk (ARMOR) model automatically detects the effective risk of software program element. ARMOR generate a complete risk model that has all the identified risks and their estimation. For the project management it shows the arithmetical measures that can be applied for risk mitigation. For the formation of risk model the consumer can use different software metrics. It also detects the source of risk and knows how to overcome these risks. In the end it shows the predicted risk of each component of the system. Regression analysis is applied for the validation of risk models. It works on statistical quantities and creates a database for risks. All the risks are assessed on the basis of risk repository.

E. *Softrisk Model*

Softrisk is another tool which is used for the risk analysis. This technique certifies the risk automation and can be applied for any type of software. This model based on the idea of risk documentation and considered that highest risk are best to save time and effort and achieving better result for overcome risk. It confirms the continuous management process till to end of the project. This model consists of eight steps.

Risk identification is the first step of this model. It can detect all type of risk that is generic and specific .The second step is estimation. In this step probability and magnitude are addressed. These are estimated by the checklist. The third step is risk documentation, which keeps all the risk data. After that risk are assisting by multiplying the magnitude and probability. Then next step is prioritize, in which risk are prioritize. At last risk the output of this is used as a input of first step.

F. *CMM based Software Risk Control Optimization Model*

Process model is used to identify the risk and their mitigation process. CMM works on the two activities. The first is software risk assessment and the second is software risk control. Software risk assessment is used to find the source of risk, their effects and prioritize them. Where as in the software risk control, risk plan are prepared and risk status are observed. Process database plays important part for risk identification and controlling decision. First of all this model detects sources of risk. Then it defines the risk parameters. After that a strategy is establish for the risk management. Then it prioritizes the risk and develops a risk mitigation process. In the end it implements the mitigation process.

IV. Comparative Analysis

In this section we have identified some risk management features that can be used to analyze the models for risk management. The table provides a comparative analysis of the techniques mentioned above in fulfilling the identified features.

Software risk evaluation(SRE) and ARMOR technique does not provide routine risk management which can cause problem for developers as the future defects will not be detected. TRM is the only technique that does not categorize the risks. The others perform risk categorization to clearly identify each and every risk and reduce them accordingly. Furthermore Riskit, Soft risk and CMMI models do not follow any predefined standards for risk management. Detecting the source of risk in very important for developer so that they can design the system accordingly but mostly the models discussed do not provide this facility. Only SRE model detects the source of risks. Risk tracking can be used to check if the risk identified has been reduced or not.

In large scale systems, the complexity, memory, bandwidth, and dependency are maximum. This increases the chances of risks in such systems. With such complex systems, risk reducing strategies are not easy to implement. Thus we have proposed a methodology for risk reduction in large scale systems. The proposed

 doi:10.20944/preprints201912.0128.v1

methodology includes all the above features of risk management. This can be done by uses a hybrid technique that covers all the features above mentioned.

In first step, the risks identified are categorized into multiple levels such as client level, requirement level, design level, execution level and maintenance level. This process will be done in the entire life cycle of the project. Furthermore there should be routine risk assessment incase a risk reoccurs. The risks tackled should be properly documented in case it reoccurs. Also there should be a proper team for risk mitigation that should communicate the risk mitigation plans to all the others involved in project development. A proper risk reducing model should be made ahead of developing so that this model is followed during the whole process. This will make it easy to analyze the process both quantitatively and qualitatively.

Models/ Technique	Identify Risk	Risk Categorization	Standard	Continuous Risk Management	Estimating Risk	Risk Documentation	Detect source of risk	Risk Tracking
SRE	Yes	Yes	Yes	No	No	No	Yes	Yes
Riskit	Yes	Yes	No	Yes	No	Yes	No	Yes
TRM	Yes	No	Yes	Yes	No	No	No	Yes
ARMOR	Yes	Yes	Yes	No	Yes	No	No	No
Soft Risk	Yes	Yes	No	Yes	Yes	Yes	No	Yes
CMM	Yes	Yes	No	Yes	Yes	No	No	Yes

V. Conclusion

During software development the most important part is the assessment, mitigation and control of risks. Thus many techniques and tools have been developed/proposed to ensure complete risk elimination. Proper risk management improves the quality of the software system. The techniques discussed above can be used for risk management as each has its own pros and cons. But above all, all of these techniques can accurately identify most of the underlying risks. This paper can help organizations understand these techniques and use the one they seem to think is the most reliable.

VI. Future Work

Future development may increase significantly and we have to deal with the types of large scale systems. With the work of the author in this field, the future large scale systems will become more secure with improved quality. Approaches that are used for large scale systems are improving as the time passes. Also the standardized infrastructures are used for the stepwise implementation of architecture.

References

[1] J. M. Bass and A. Haxby, “Tailoring Product Ownership in Large-Scale Agile Projects: Managing Scale, Distance, and Governance,” IEEE Software, vol. 36, no. 2, pp. 58–63, 2019.

[2] J. Grossmann, M. Felderer, J. Viehmann, and I. Schieferdecker, “A Taxonomy to Assess and Tailor Risk-Based Testing in Recent Testing Standards,” IEEE Software, pp. 1–1, 2019.

[3] W.-M. Han, “Discriminating risky software project using neural networks,” Computer Standards & Interfaces, vol. 40, pp. 15–22, 2015.

[4] M. Keil, A. Rai, and S. Liu, “How user risk and requirements risk moderate the effects of formal and informal control on the process performance of IT projects,” European Journal of Information Systems, vol. 22, no. 6, pp. 650–672, 2013.

[5] S. Liu and L. Wang, “Understanding the impact of risks on performance in internal and outsourced information technology projects: The role of strategic importance,” International Journal of Project Management, vol. 32, no. 8, pp. 1494–1510, 2014.

[6] M. Pasha, G. Qaiser, and U. Pasha, “A Critical Analysis of Software Risk Management Techniques in Large Scale Systems,” IEEE Access, vol. 6, pp. 12412–12424, 2018.

[7] K. Sangaiah, O. W. Samuel, X. Li, M. Abdel-Basset, and H. Wang, “Towards an efficient risk assessment in software projects–Fuzzy reinforcement paradigm,” Computers & Electrical Engineering, vol. 71, pp. 833–846, 2018.

[8] Y. Takagi, O. Mizuno, and T. Kikuno, “An Empirical Approach to Characterizing Risky Software Projects Based

on Logistic Regression Analysis," Empirical Software Engineering, vol. 10, no. 4, pp. 495–515, 2005.

[9] Y. Zhou and J. Yan, "A Logistic Regression Based Approach for Software Test Management," 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016.

[10] A. E. Yamami, S. Ahriz, K. Mansouri, M. Qbadou, and E. H. Illoussamen, "Rethinking IT project financial risk prediction using reference class forecasting technique," 2018 4th International Conference on Optimization and Applications (ICOA), 2018.

[11] P. Shah, N. Davendralingam, and D. A. Delaurentis, "A conditional value-at-risk approach to risk management in system-of-systems architectures," 2015 10th System of Systems Engineering Conference (SoSE), 2015.

[12] D. Subramanian, D. Bhattachrajya, R. R. Torrado, J. Kephart, V. Chenthamarakshan, and J. Rios, "A cognitive assistant for risk identification and modeling," 2017 IEEE International Conference on Big Data (Big Data), 2017.

[13] H.-C. Liu, L.-E. Wang, Z. Li, and Y.-P. Hu, "Improving Risk Evaluation in FMEA With Cloud Model and Hierarchical TOPSIS Method," IEEE Transactions on Fuzzy Systems, vol. 27, no. 1, pp. 84–95, 2019.

[14] M. Bahroun and S. Harbi, "Risk management in the modern retail supply chain: Lessons from a case study and literature review," 2015 International Conference on Industrial Engineering and Systems Management (IESM), 2015.

[15] J. B. Oliveira, R. S. Lima, J. E. Kobza, and M. Jin, "An analysis on logistics risk management: Tools, techniques and review," 2016 6th International Conference on Information Communication and Management (ICICM), 2016.

[16] N. Mathuthu, A. Marnewick, and H. Nel, "A review of risk management techniques and challenges in harbour and port expansions," 2017 Ieee Africon, 2017.

[17] I. Gunawan, T. Nguyen, and L. Hallo, "A Review of Methods, Tools and Techniques Used for Risk Management in Transport Infrastructure Projects," 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2018.

[18] T. Mikkonen and A. Taivalsaari, "Software Reuse in the Era of Opportunistic Design," IEEE Software, vol. 36, no. 3, pp. 105–111, 2019.

[19] A. Elzamly, B. Hussin, and N. M. Salleh, "Top Fifty Software Risk Factors and the Best Thirty Risk Management Techniques in Software Development Lifecycle for Successful Software Projects," International Journal of Hybrid Information Technology, vol. 9, no. 6, pp. 11–32, 2016.

[20] "The Methodology of Risk Analysis in Assessing Information Security Threats," Modeling of Artificial Intelligence, vol. 4, no. 2, May 2017.