International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-10, Issue-2, March 2022
https://doi.org/10.55524/ijircst.2022.10.2.36
Article ID IRPV1057, Pages 179-184
www.ijircst.org

Software Risk Management Approaches in Large-Scale Systems: A Critical Examination

Sushil Bhardwaj

Assistant Professor, Department of Computer Applications, RIMT University, Mandi Gobindgarh, Punjab, India

Correspondence should be addressed to Sushil Bhardwaj; sushilbhardwaj@rimt.ac.in

Copyright © 2022 Made Sushil Bhardwaj. This is an open-access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT- For a long time, software industry researchers have worked on risk supervision solutions. Software risk supervision is a computer engineering technique that includes identifying risks, estimating risks, mitigating risks, and monitoring them. It provides a structured framework in which to make informed decisions about software development issues. Because of its complexity, measuring risks in a large-scale system is more challenging. Large-scale systems are difficult to build since numerous hazards may emerge throughout the process. Risk factors in large-scale systems vary from those in small systems, particularly in terms of independent components. This article explains the distinction among high-level and low-level system, as well as a comprehensive list of risk variables. The tools from the literature are further classified into subcategories based on their suitability. We provide a thorough comparison study of several software associated risk supervision replicas with certain frequently recognized characteristics, and then classify them depending on the strictness of the respective hazards.

KEYWORDS- Risk supervision, High-level System, Risk Issues, Risk Supervision Tool.

I. INTRODUCTION

Risk is an issue that may result in significant losses and put different organizational processes at risk. Networks, the Internet, malicious programs or users, vulnerabilities, and physical security may all pose dangers in computer science areas [1,2]. There are many hazards involved in developing great software systems with better quality. Although these risks might not be fully avoided, plan directors may minimize their effect on products by assessing the risks associated with IT resources. Software systems have grown increasingly sophisticated in recent years, and they are now referred to as large-scale systems. As the scale and complexity of a project grows, so do the risks associated with it. The software industry is one of the world's most important industries [3-7]. According to, off-the-shelf computer program is retailed for an average of $350 billion each year. As of the great superiority and trustworthiness expectations, risk supervision is a speculation that may be beneficial in the future for large-scale systems.

The primary aim of a risk management system is to detect and handle any potential hazards during software development before they arise [8]. Risk is often divided into two categories: systematic risks and unsystematic hazards. External causes like as hacking, viruses, natural catastrophes, and power outages are examples of systemic hazards. A susceptible browser is an example of a systemic risk, wherein any type of flaw could be responsible for security breaches and damage an organization's resources. Unsystematic risks, such as the abuse of sensitive records, app errors, insider assaults, information loss, apparatus failures, and human interactions, all provide distinct hazards to the business. Furthermore, methodical perils are referred to as generic risks, while unsystematic hazards are referred to as particular risks. The categorization of hazards in software development is shown in Figure 1. A risk management team in software development constantly analyzes and monitors different risks and determines their negative effects on software development.

By prioritizing and rating risks, risk management systems offer a dynamic method to make decisions. A peril supervision scheme is often built on the identification and valuation of hazards. Software development risk cannot be eradicated, as it cannot be removed in many other companies, but risk managers may minimize the effect of these risks by using proper risk management tools and methods.

A. Risk Supervision Utilities

The risk supervision progression necessitates the use of technologies that can detect and mitigate perceived threats [9,10]. We have identified and classified several risk management solutions that are regarded to be useful for reducing risks based on an analysis of the current literature on risk management in software development.

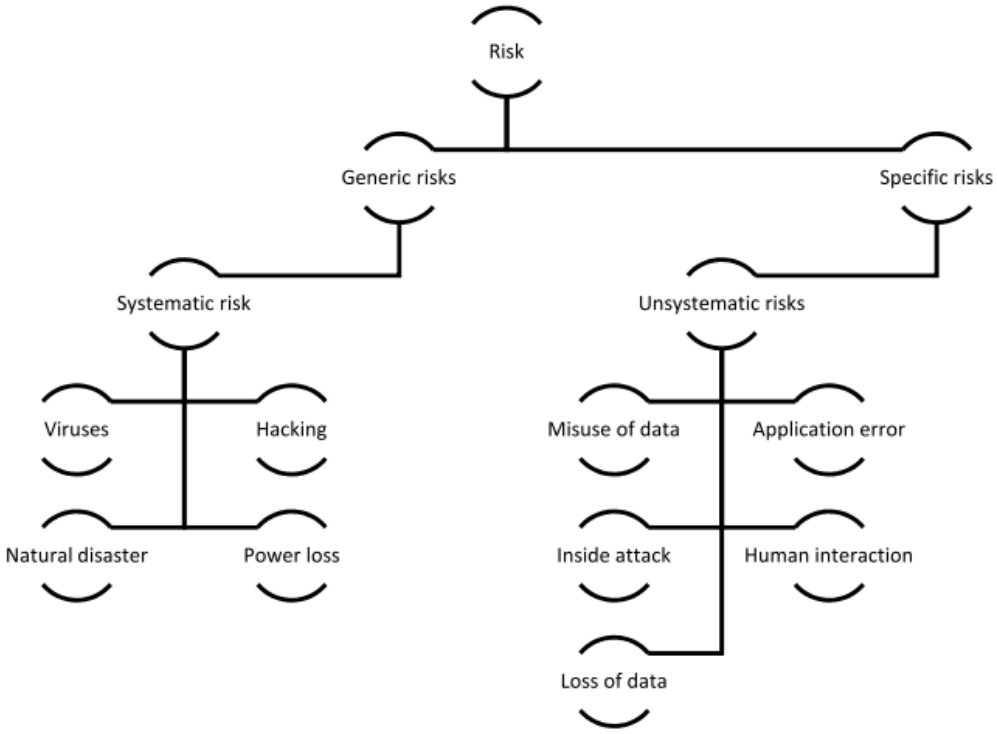

Figure 1: Sorts of hazards in computer program development [1]

II. BACKGROUND

In 1980, the risk management method was made explicit in the software development process. In the field of software engineering, Boehm Barry is regarded as the "Creator of Risk Supervision" [11]. He developed the risk-driven spiral model and stressed the need of risk management in software development. An increase in the demand for software initiatives in recent years has increased the dangers [12-16]. Some software problems grow increasingly complicated as programming languages progress, making risk management harder in such projects. As a consequence, this risk management method is the subject of much study. The goal of the study is to enhance the area of software management by improving software management methods.

The top ten software hazards from various research, as well as their risk factors, were evaluated and collected in the available literature. According to the author's findings, certain risks exist in software development processes in the areas of planning and controlling. The growing importance of peril supervision in computer software development to provision and prevent hazards is discussed in this reference. The authors provide a information model for hazard detection and evaluation. Reference delves into the areas of risk supervision and data administration. The authors recommend KBRM (Knowledge Based Risk Management), a conceptual framework that combines KM and RM to improve the performance of IT projects [17]. The research identifies several critical aspects for the KBRM structure and suggests few methods for better risk supervision. Orientation outlines a risk supervision strategy aimed only for small companies. The research is based on a thorough examination of several small company software development options. The study offers a list of risk management tools as well as some risk reduction methods. The connection between project success and risk management is described in reference. The soft and hard elements of risk management systems are also combined with project success in this approach. Interviews with risk and project managers were conducted in order to examine the hard and soft elements of risk management systems for project success. Reference provides examples of risk measurement using qualitative and quantitative data from a variety of real-world industrial situations. This study examines the risks' dimensions and sources, as well as how to apply theoretical methods to businesses with plan directors. The reference provides a collection of methods culled from current literature, as well as some recommendations for their use in real-world situations [18-22]. This research aids in the integration and effective operation of risk management and knowledge management. A computer program structure for RRA (Rapid Risk Assessment) in incorporated resource chain risk supervision is shown in Reference. To access and prioritize hazards, the suggested approach integrates qualitative and quantitative data. Quantitative data is based on fuzzy logic and probability theory, whereas qualitative data is based on surveys. A study of the risk management system is included in this reference, as well as a discussion of various risk reduction techniques. The lessons also include a study of risk supervision replicas as well as methods for selecting the optimal risk assessment methodology among them.

A. Computer Program Risk Supervision in High Level System

Social fault, program execution catastrophe, hardware catastrophe, and managerial mistake are only some of the hazards involved with software development [23]. In general, there are many risk management approaches and methodologies used in designing computer programs. Computer program initiatives have a higher failure rate than other companies, yet it requires technical and costly resources, much like other firms. Furthermore, as EW grows in scope and intricacy, risk extenuation in App designing becomes increasingly difficult.

The unmanageability and uncontrollability of large-scale system development is a significant issue. The major problems, according to, are a lack of appropriate standards and recurrent fluctuations in the criteria. High grade systems are comprised of human networks and technological resources such as machinery, computers, and software. The interconnections, interdependencies, and attention given to the components distinguish large-scale and small-scale systems. Researchers have emphasized on dependency, risk migration, and lengthy incubation periods when it comes to risk reduction in large-scale systems. Large-scale systems, in general, are more challenging to manage due to their complexity. In order to function properly, the big system is split into subcomponents and subfactors. Risk reduction is achieved by splitting the system into smaller groups.

B. Risk Valuation Outline

Large-scale systems are difficult to understand. For a good end result, they need meticulous processing, planning, and execution. Risk is a critical problem in software development, and software development suffers from a lack of effective risk mitigation [24]. After the risk identification process, risk managers identify all potential risk variables, which is the first step towards risk reduction. After that, the risks will be classified and prioritized. Managers and senior executives evaluate the risks' effect on software development. The most significant hazards will be addressed first, using this risk estimate. This procedure is advantageous in reducing large-scale organizational losses. The high-impact hazards will next be examined. Risk managers will develop strategies for mitigating risk and implementing risk management methods. Risk managers monitor and report on the process once these countermeasures are implemented. This procedure will begin again with the risk identification phase if additional gaps are discovered. The team members should manage and track the risk during the monitoring and reporting phase. The software project's stakeholders assess the risk input in the last phase. They will keep the team up to date and offer their input, as well as keep track of the process.

C. Risk Factors in Software Development

This research presents a comprehensive list of risk variables culled from the available literature. After these variables have been identified, they are classified into several stages, including the user level phase, requirement collecting, planning, analysis, design, implementation, and maintenance.

D. Small-Scale Vs Large-Scale Software

The hazards may impact small, medium, and big software development companies. The first step in risk assessment is to identify the issue and its cause [25]. Risk assessment for large-scale systems is a problem in and of itself, since large-scale software systems, according to, are like a riddle. Because large-scale systems have more complicated and independent components, risk management is more challenging. Many software projects provide the appropriate functionality and performance that the creators promised, while other software projects fail to meet the needs of consumers within the time and budget constraints. Risk may arise in every project, compromising the end product's functioning. Risk managers should take necessary actions to reduce or prevent hazards. The failure of a large-scale system may impact the stakeholders of that product if appropriate risk management methods are not used. Customers', workers', and organization's budgets and time may be squandered if the end product fails. The business's earnings may be harmed as a result of the failure.

III. DISCUSSION

The size of software development systems has been steadily growing in recent years. Intricacy, lines of code, bandwidth, memory, communiqué, asset, dataset, dependence, and many other variables of a system are growing by the day, making risk mitigation harder in these systems. Some methods for peril reduction in a high-level software development systems are given in this approach. Risk variables such as risks of operator level, prerequisite level risks, preparation, examination, strategy, enactment, and conservation are all categorizes in the suggested approach. The risk mitigation in large-scale systems is shown in Figure 2.

A. Define Rules and Policies:

The organization establishes certain rules and policies at the start of this approach. The regulations and policies will be incorporated in this phase, and the formal work will be completed.

B. Roles and Responsibilities Assignments

Qualified and competent individuals will be given roles. This procedure will be handled by senior management, who will appoint an experienced manager to the project who will be able to oversee the whole software development process. Many workers are engaged in large-scale systems, and managing them is tough and hard. As a result, tasks and responsibilities will be allocated to the appropriate people early on.

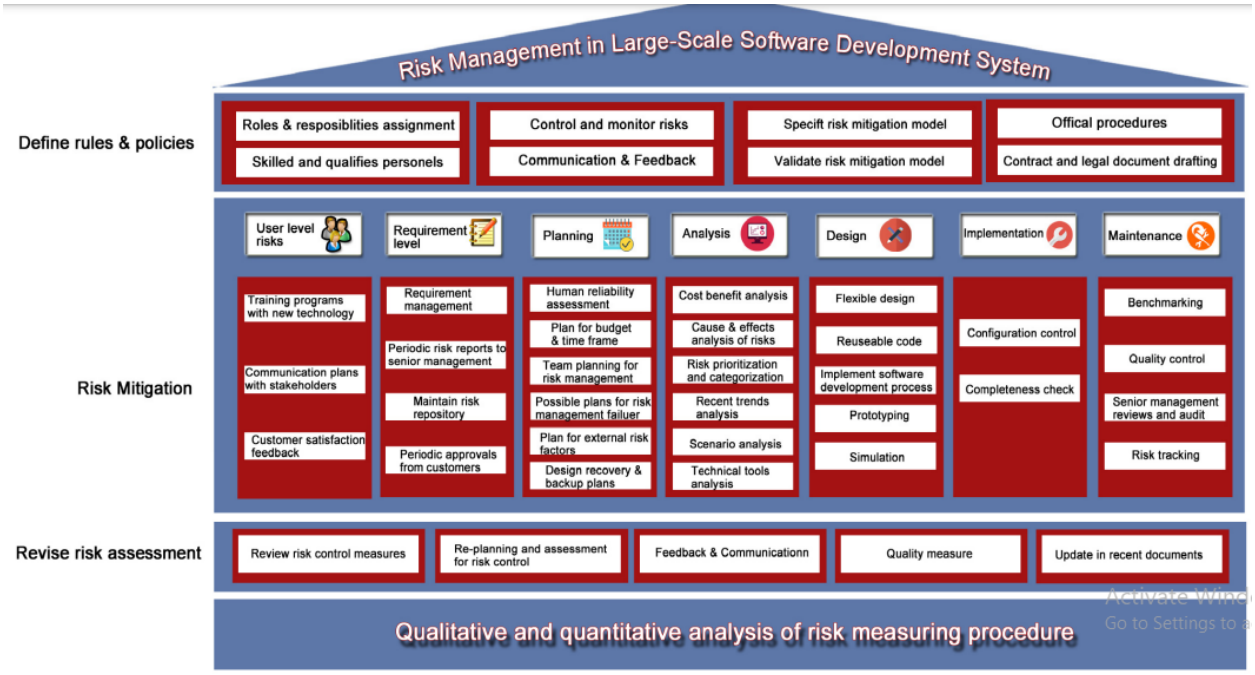

Figure 2: Proposed methodology for risk mitigation in large-scale systems

1) Observation-based Control Risks

This phase will create the configuration that shall observe and interact through all group participants involved in the risk mitigation process. Plans for communicating with senior management and receiving input will be made here.

2) Official Procedures

All formal work will be completed at this phase. Investors and project partners will write and sign contracts and other legal papers. This is a critical stage in large-scale systems before moving on to risk reduction.

C. Risk Mitigation

The previously mentioned risk mitigation method in software development is classified in this second phase as per the operator level, prerequisite level, scheduling, analysis, design, implementation, and maintenance level.

1) User Level Risks

User-level risks are those that arise from the user's perspective, such as minimal user engagement and a lack of understanding of newly created technology. The business will educate users on the new technology to avoid these kinds of dangers. With the help of all stakeholders, communication strategies will be developed to assist the business in interacting with users. Customer satisfaction comments will be collected at the last stage. This phase will be repeated throughout the development process.

2) Requirement Level

Users or consumers may be unable to communicate their demands to developers, and frequent changes in requirements may be hazardous to the business. To deal with these kinds of risks, the company will need to use requirement management. Risk reports will be produced on a regular basis and sent to senior management. The requirements and any modifications made by the users will be overseen by the senior management. The company will keep track of all potential hazards in a risk repository, which will aid in the organization's ability to prevent future risks. Customers' comments will be evaluated on a regular basis by the company. This process may be beneficial to both the consumer and the business.

D. Revise Risk Assessment

In the event that a risk recurs, a revised risk assessment will be performed following risk reduction. This creates a continual risk management process evaluation.

1) Review Risk Control Measures

The reviews of the risk management process will be produced in this phase. This approach will become a continual risk management procedure as it is reassessed on a regular basis. The system becomes more measurable for risk management when the process is revised.

2) Assessment & Re-planning

If there is a danger of recurrence, the project will go through the re-planning process.

3) Feedback and Communication

Feedback and contact with all stakeholders and senior management will be organized at this point. This phase will need reapplying any modifications made throughout the re-planning process.

4) Quality Measure

The ability to assess the finished product's quality is critical for both organizational and consumer satisfaction.

5) Update in Recent Documentation

The paperwork procedure will be completed last. The most current papers will also be updated to reflect recent developments and trends.

E. Quantitative and Qualitative Examination Of Peril Determining Technique

The whole risk management process will be subjected to qualitative and quantitative analysis in the final phase. Graphical charts and presentations will be created for qualitative assessments. Top management will conduct surveys and interviews for quantitative assessments.

Software risk management is a growing field with the primary goal of identifying, assessing, and reducing perceived risks. To detect and control risks, software risk management models are employed. Models vary in terms of characteristics. We use several common characteristics from the literature to compare and contrast these models in our research.

The Software Risk Evaluation (SRE) is a risk management method that gives a more detailed overview. This method is thought to be effective for identifying and analyzing hazards. The SRE is utilized as a decision-making tool in addition to being useful for risk management. The classification of hazards offers a clear and comprehensible picture of dangers. It is based on SEI (Software Engineering Institute) standards and covers risk identification, risk analysis, planning, and communication as well as other fundamental components. It keeps track of hazards in a methodical way by capturing pictures of them. In the risk mitigation process, the SRE creates a shared perspective and a common framework for all team members. This method includes members of the team, but not all stakeholders. Stakeholder engagement is lacking, which makes the process hazier and perhaps leads to failure.

TRM (Team Risk Management) depicts the risk management process' operational operations and organizational structure. It controls risks throughout the software development process with the participation of all people and stakeholders, making decision-making simpler and more efficient. It also adheres to SEI guidelines and integrates all stakeholders (including investors, managers, developers, and consumers) into the company. Because hazards may occur at any stage of software development, TRM regularly and collaboratively guarantees the ongoing risk management process throughout software development. Managers establish the planned tasks and perform them in a continuous cycle in this risk management approach. Risks may be detected and evaluated on a regular basis using TRM. With frequent evaluations and constant monitoring of the risk mitigation process, it maintains track of hazards.

For risk reduction, the Capability Maturity Model (CMM)-based model follows the CMM framework. A database is utilized to detect and assess hazards in this approach. This approach combines risk evaluation and risk management. Risk assessment is the initial stage in identifying and prioritizing risks. Risk mitigation strategies are developed in the second stage of risk control, and risk monitoring is done to update existing hazards and track future risks. This is a well-organized risk management framework that will serve as a central repository for detecting and managing risk. This is an important stage in risk management decision-making.

IV. CONCLUSION

The evaluation and management of risk variables is the most important aspect of software development. In recent decades, several risk management methods and accompanying technologies have been presented. Organizations have been compelled to adopt these methods to enhance product quality and risk assessment due to intense competition. As a result, the goal of this research is to offer a critical examination of the various risk management models using certain characteristics gleaned from the literature. This research helps others in selecting the appropriate model or framework for their business. This study also provides a comprehensive list of risk variables as well as risk management strategies. An examination of the selected risk management models from the available literature will be aided by a comparison analysis. Some common characteristics and variables are identified, and the models are evaluated using these. We may infer from this comparison that any model with various parameters may be appropriate in different situations.

REFERENCES

[1] Pasha M, Qaiser G, Pasha U. A critical analysis of software risk management techniques in large scale systems. IEEE Access. 2018;

[2] Thom-Manuel OM, Ugwu C, Onyejegbu LN. A New Mathematical Risk Management Model for Agile Software Development Methodologies. Int J Softw Eng Appl. 2018;

[3] Ghai W, Kumar S, Athavale VA. Using gaussian mixtures on triphone acoustic modelling-based punjabi continuous speech recognition. In: Advances in Intelligent Systems and Computing. 2021.

[4] Khatri M, Kumar A. Stability Inspection of Isolated Hydro Power Plant with Cuttlefish Algorithm. In: 2020 International Conference on Decision Aid Sciences and Application, DASA 2020. 2020.

[5] Sharma K, Goswami L. RFID based Smart Railway Pantograph Control in a Different Phase of Power Line. In: Proceedings of the 2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020. 2020.

[6] Goswami L, Kaushik MK, Sikka R, Anand V, Prasad Sharma K, Singh Solanki M. IOT Based Fault Detection of Underground Cables through Node MCU Module. In: 2020 International Conference on Computer Science, Engineering and Applications, ICCSEA 2020. 2020.

[7] Solanki MS, Sharma DKP, Goswami L, Sikka R, Anand V. Automatic Identification of Temples in Digital Images through Scale Invariant Feature Transform. In: 2020 International Conference on Computer Science, Engineering and Applications, ICCSEA 2020. 2020.

[8] Tumbinskaya M V. The use of intelligent systems for risk management in software projects. Mordovia Univ Bull. 2017;

[9] Chandrinos SK, Sakkas G, Lagaros ND. AIRMS: A risk management tool using machine learning. Expert Syst Appl. 2018;

[10] Ghaeli MR. The advantage of project risk management tools. J Proj Manag. 2018;

[11] Teklemariam MA, Mnkandla E. Software project risk management practice in Ethiopia. Electron J Inf Syst Dev Ctries. 2017;

[12] Kumar A, Jain A. Image smog restoration using oblique gradient profile prior and energy minimization. Front Comput Sci. 2021;
[13] Gupta N, Vaisla KS, Jain A, Kumar A, Kumar R. Performance Analysis of AODV Routing for Wireless Sensor Network in FPGA Hardware. Comput Syst Sci Eng. 2021;
[14] Gupta N, Jain A, Vaisla KS, Kumar A, Kumar R. Performance analysis of DSDV and OLSR wireless sensor network routing protocols using FPGA hardware and machine learning. Multimed Tools Appl. 2021;
[15] Gupta B, Gola KK, Dhingra M. HEPSO: an efficient sensor node redeployment strategy based on hybrid optimization algorithm in UWASN. Wirel Networks. 2021;
[16] Kumar Gola K, Chaurasia N, Gupta B, Singh Niranjan D. Sea lion optimization algorithm based node deployment strategy in underwater acoustic sensor network. Int J Commun Syst. 2021;
[17] Neves SM, da Silva CES, Salomon VAP, da Silva AF, Sotomonte BEP. Risk management in software projects through Knowledge Management techniques: Cases in Brazilian Incubated Technology-Based Firms. Int J Proj Manag. 2014;
[18] Khanna R, Verma S, Biswas R, Singh JB. Implementation of branch delay in Superscalar processors by reducing branch penalties. In: 2010 IEEE 2nd International Advance Computing Conference, IACC 2010. 2010.
[19] Gupta H, Kumar S, Yadav D, Verma OP, Sharma TK, Ahn CW, et al. Data analytics and mathematical modeling for simulating the dynamics of COVID-19 epidemic—a case study of India. Electron. 2021;
[20] Gupta H, Varshney H, Sharma TK, Pachauri N, Verma OP. Comparative performance analysis of quantum machine learning with deep learning for diabetes prediction. Complex Intell Syst. 2021;
[21] Sharma TK. Enhanced butterfly optimization algorithm for reliability optimization problems. J Ambient Intell Humaniz Comput. 2021;
[22] Hirawat A, Taterh S, Sharma TK. A dynamic window-size based segmentation technique to detect driver entry and exit from a car. J King Saud Univ - Comput Inf Sci. 2021;
[23] Darabseh A, Al-Ayyoub M, Jararweh Y, Benkhelifa E, Vouk M, Rindos A. A novel framework for software defined based secure storage systems. Simul Model Pract Theory. 2017;
[24] Bernard L. A Risk Assessment Framework for Evaluating Software-as-a-Service (SaaS) Cloud Services Before Adoption. ProQuest Diss Theses. 2011;
[25] Šmite D, Moe NB, Šāblis A, Wohlin C. Software teams and their knowledge networks in large-scale software development. Inf Softw Technol. 2017;