Journal of Intelligent System and Applied Data Science (JISADS)

Journal homepage : https://www.jisads.com
ISSN (2974-9840) Online

SOFTWARE PROJECT FAILURE AVOIDING THROUGH RISK ANALYSIS AND MANAGEMENT

Mohammad Ibraigheeth1[*1]*

[1]*Department of Software Engineering, Bethlehem University, Bethlehem, Palestine, mayyash2010@gmail.com*

ABSTRACT

Software project failures can have undesirable effects, including financial loss, operational disruptions, and compromised safety. To address these challenges, effective risk analysis and management are essential. This paper presents a review of the literature on software project risk management, focusing on various aspects crucial for avoiding project failures. It begins with an exploration of risk classification systems, highlighting how categorizing risks can aid in better understanding and managing them. The paper underscores that classifying risks based on their duration, impact, and source can significantly improve the effectiveness of risk management strategies. A systematic approach, including identification, classification, analysis, planning, tracking, control, and communication, offers a robust framework for mitigating potential threats and minimizing their impact. Various risk response strategies, such as avoidance, transfer, reduction, and acceptance, provide diverse methods for managing risks depending on their nature and severity. Additionally, it addresses the importance of aligning risk management practices with established standards, specifically the IEEE Software Failure Standards, to ensure compliance with industry benchmarks and enhance the reliability of these practices. In conclusion, effective risk management is fundamental to the success of software projects. Through a structured approach to risk assessment and the application of appropriate response strategies, organizations can navigate uncertainties more effectively, improve project outcomes, and achieve their objectives with greater confidence.

Keywords: Risk analysis, risk assessment process, risk response strategies, software failure, software failure standards.

1. INTRODUCTION

Assessing and addressing software project risks early in the development process can mitigate the effects of undesirable events that could lead to project failure [1] is prone to multiple threats through the advancement and application of software. Generally, there are three types of software risks:

First, the failure of a software project as a business results in wasted money and time, as well as a lost business opportunity. This type of risk is known as software project risk (including software development risks and IT project risks).

Second, there is the threat to the safety of citizens and the environment. Failure of the software system may result in an accident that, in the worst case, could lead to loss of life. This is known as safety software risk.

Third, the system's service may deteriorate, or the information system resources may be compromised or negatively manipulated if the integrity of the system is violated through malicious activities by an attacker. This is known as security software risk.

In spite of advancements in technology, software projects still encounter many problems. Customer requirements are often not deeply understood, resulting in continuous expansion of the system scope or even final system rejection. Human involvement introduces factors such as personality and cognitive biases into the technical challenges of projects. Additionally, software programs are prone to errors, and cooperation among project members is frequently weak. Consequently, customer expectations are often unmet. These issues indicate a need for significant improvements in software development and procurement.

One of the most influential approaches recognized in all software engineering and project management manuals [2~7]

To understand a risk thoroughly, it is necessary to obtain a detailed description so that a common understanding of the risks can be achieved, and ownership and responsibilities can be clearly defined. The following are examples of information that could be recorded to fully understand a risk [8]:

- Name of risk
- Scope of risk, including events and related dependencies
- Nature of risk
- Stakeholders
- Risk tolerance, attitude, and appetite
- Events' probabilities and magnitudes
- Standards and mechanisms needed
- Developing a risk management strategy
- Responsibility
- Scheduling risk improvements

The above list of information could be applicable to hazardous risks, and the list should be modified to provide a full description of control or opportunity risks so that the correct range of information about each risk can be collected.

Software project failures often result in severe consequences, including financial loss, operational disruptions, and compromised safety. To mitigate these risks, it is crucial to develop and implement effective risk management strategies. This paper seeks to explore and address the challenges associated with managing risks in software projects. The primary objectives of this research are to investigate how risk classification systems, risk assessment processes, and response strategies contribute to the prevention of project failures. Specifically, this paper aims to answer the following research questions:

How can risk classification systems enhance the clarity and effectiveness of risk management strategies?

What systematic approaches to risk management are most effective in mitigating potential threats?

How can aligning risk management practices with established software failure standards improve project outcomes?

The paper is structured as follows: Section 2 delves into Risk Classification Systems, exploring methods to categorize and understand risks. Section 3 covers Risk Management, presenting a comprehensive framework for handling potential threats. Section 4 discusses the Risk Assessment Process, detailing key stages such as identification, classification, analysis, planning, tracking, control, and communication. Section 5 examines various Risk Response Strategies, evaluating their effectiveness in different scenarios. Section 6 addresses the importance of aligning risk management practices with Software Failure Standards, particularly the IEEE criteria. Finally, Section 7 concludes the paper by summarizing the findings and emphasizing the importance of systematic risk management in ensuring the success of software projects.

2. RISK CLASSIFICATION SYSTEMS

Many features are considered when classifying risks; the most important are the duration of their effect and the consequences of that effect. Another feature to consider in classifying risks is the source of the risk, where the origin, such as counterparty or credit risk, is the basic scale for classification.

Taking into account the nature of the risk's effect is another strategy for classifying risks. Some risks might severely affect the organization's financial income, while others might impact infrastructure and organizational interests. More dangerously, risks might negatively affect the organization’s reputation and its competitive environment.

Higher authorities in an organization usually identify the nature of the risks facing their organization and then decide on the best risk classification strategy to adopt, considering the organization's activities and duties. It is noteworthy that certain risk classification frameworks are adopted by management, which obliges the organization to strictly follow the procedures assigned under each framework.

Any chosen risk management system must be fully compatible with the nature of the organization because no single universal system is applicable to all types of organizations. It might be possible that many strategies could be utilized to classify risks to obtain a better and clearer understanding of what the organization is really facing.

While not a formal system, it does not deny the fact that short, medium, and long-term risk classification significantly promotes identifying potential risks because they are basically and respectively connected to the organization's activities, plans, and strategies. This distinction might not be a final decisive factor in identifying risks, but it surely contributes to a more advanced risk classification. This does not guarantee that some short, medium, and long-term risks will not happen, which will consequently affect the basic operational process.

The effect of short-term risks can be immediately noticed on the organization's aims, basic dependencies, and fundamental procedures. The danger of these risks lies in disrupting operations on the spot. Although it is not a prevailing case, short-term risks are principally hazardous. The main cause of these risks is usually attributed to poorly planned events that might be disruptive. These short-term risks have a substantial negative effect on the main processes of the organization, which consequently badly affects the sustainability of routine procedures.

Unlike the immediate impact of short-term risks, the impact of medium-term risks might be effective months or a year later. It is generally accepted that this type of risk affects the organization's ability to maintain the effective basic operations responsible for managing tactics, projects, improvements, and product releases.

Compared to short and medium-term risks, the effect of long-term risks might be felt after more than five years. This type of risk is particularly associated with hindering the organization’s ability to ensure the continuity of basic processes responsible for implementing influential strategies. Although this type of risk mainly targets strategy, it should not be viewed as particularly related to opportunity management. Since this type is capable of eroding the foundation of the organization, it is capable of destroying more values and principles.

3. RISK MANAGEMENT

Organizations are increasingly aware of the benefits that explicit risk management brings. By adopting proactive risk management strategies, several improvements can be expected. With identified disruptive actions and assigned strategies to overcome them, organizations can maintain more effective processes and contain the harmful effects of disruption, ultimately leading to cost reduction. Higher management will be able to determine the best processes for activities and become aware of available alternatives if the organization is exposed to certain types of risks. These measures will positively impact projects.

Proactive strategies enable management to develop an effective strategy where risks are thoroughly investigated, and strategic decisions are made adequately. This guarantees that the newly developed strategy will achieve the desired outcomes. It is intolerable for organizations to suffer financially, have their operations disrupted, distort their reputation, or lose their competitive markets due to unexpected events. Stakeholders expect organizations to take all necessary steps to ensure the smooth and unhindered delivery of projects.

The primary objective of risk management is to ensure project success through clear and effective treatment of future uncertainties. It also aims to conduct an adequate and trusted evaluation of risks and strive to reduce their disruptive consequences. A pioneer in risk management, believes that effective risk management can reduce about 40% of the cost of software projects when work is well-managed [9].

Risk assessment is best described as a systematic strategy to identify and analyze risks that any given project might be exposed to. Effective risk assessment requires a thorough review of risk reports and the reuse of gathered experiences in facing risks. An adequate evaluation of risks contributes significantly to avoiding common risks and becoming familiar with potential future risks. Other tools that provide data about risks can also be helpful for evaluating risks.

Hazard risks actively prevent organizations from achieving their desired objectives. These risks are closely associated with insurance-related issues such as fire, damage, and theft. The risk management system is characterized by its deep roots in managing and controlling hazardous risks. Activities with normal efficiency might be disrupted due to loss, theft, or damage, affecting people, information technology (IT), suppliers, assets, premises, and communications [10].

Control risks generate uncertainty regarding the achievement of the organization's objectives. An example of control risks is better seen in the protocols assigned to control internal finance. Removing control protocols eliminates the ability to anticipate future events. Although it is difficult to give an exact description of control risks, further illustrations will help understand them. Uncertainty is the prevailing characteristic of control risks, such as noncompliance with legal instructions and significant losses due to fraud. These risks are often based on two main factors: the successful management of individuals and the proper utilization of control protocols. Despite organizations' efforts to manage control risks carefully, these risks still pose significant threats.

Opportunity risks are those that organizations usually and intentionally seek. These risks generally arise from organizations' attempts to extend their objective realization but might hinder progress if adverse results occur. Organizations view opportunity risks as the most promising for long-term success. Investing in high-risk deals can be tempting for organizations since high risk is associated with high profit. However, not all organizations are willing to invest their most valuable resources in hard, risky, and unguaranteed ventures

Theoretically speaking, from an organizational perspective, risks emerge when organizations exert efforts to overcome the issue of uncertainty, driven by the determinants of cost and capability. The difficulty lies in finding a position on these areas that would clarify a risk record accepted by stakeholders. Thus, risk and its management can be seen as a strategic question subject to compromise. A risk-averse strategy might not achieve outstanding success; however, a strategy based on embracing risks is likely to increase losses. Explicitly managing this balance is often marginalized in favor of pursuing the desired mission [11]

Regarding projects, software projects have always been considered high-risk ventures that might fail [12]. Project risks can be classified into two categories: generic risks, which are widespread among projects, and project-specific risks [13]. Many of these risks are manageable and identifiable, but others are more difficult to control, and their impact is unpredictable. This is particularly troublesome when a project has multiple dimensions, such as size, structure, complexity, composition, context, novelty, long planning and execution horizons, and volatile change [14]. However, the importance of management in software projects is evident in avoiding fatal problems, preventing reproduction, keeping efforts focused and concentrated, and elevating the level of win-win situations [15]. While software projects are not always the source of risks, these risks can significantly impact outcomes.

Risk and its management are crucial since IT projects can act as a means to facilitate organizational change that supports IT. Consequently, the success of work is highly dependent on the success of managing risks

4. RISK ASSESSMENT PROCESS

4.1 Risk identification

In the process of risk management, identifying the risk is the primary step to be taken. When risks are successfully identified, they are listed under a known-risks list. It is significantly important to identify risks early because the management can address them before further complications arise [16]. If this step is successful, then all risks that threaten the success of the project will be detected early. Identifying risks can be accomplished through various channels, such as interviewing customers and vendors.

Using open-ended questions is a fruitful strategy for identifying likely risk areas. Voluntary reporting is also effective, especially when higher management offers rewards and privileges to those who identify risks and bring them to management's attention. Of course, this strategy requires the absolute removal of the "shoot the messenger" mentality. Breaking down existing structures is another effective strategy for identifying risk areas. Additionally, classifying risks according to problems that occurred in other projects can be helpful as a record for investigating new emerging risks [17].

4.2 Risk classification:

Risk classification is important in providing a framework to organize and investigate the problems that might arise during the process of developing software [18]. It forms the foundation for identifying and organizing the complete set of software development risks, whether they are technical or non-technical. Another method of classification involves determining the domain of influence, as proposed by Tiwana and Keil [20]. They believe that project managers can identify risks that are either within their fields or that come from external sources. Consequently, they tend to classify risks into two areas: the project manager's domain and the customer's domain

4.3 Risk analysis

Risk analysis is the process of converting the data provided and collected about a certain risk into a decision. Analyzing risks enables the project managers to decide on which risk to work and how to work on it [20]. The process of analyzing risks, every single risk is deeply investigated to figure out: Probability: the possibility that the risk will lead to loss and Impact: The amount of loss if that risk grows to be a problem.

The Risk Exposure is defined to assist identifying risks' priorities qualitatively. Risk Exposure is meant to assert the effect that takes place due to a risk regarding the amount of loss. Risk Exposure (RE) is best described as the possibility of undesired results which might be obtained and which increase the amount of loss [21].

*RE = Probability of unexpected outcome * Loss of unexpected outcome* *(1)*

The Risks list is arranged in priority according to the outcomes of the risk analysis. Because source restrictions hardly permit all risks considerations, risks that require planning and extra work are prioritized. Other risk might be postponed for future investigations. Due to certain changes in the work environment, prioritized risks are subjected to periodic revision [17].

4.4 Planning

Planning is the process in which risk information are converted to be decisions and actions. Planning is also viewed as the process of developing certain procedure to handle individual risks, identifying the priorities of risk actions, and creating a complete plan for risk management [22]. Risk management plan might be formed based on different strategies such as [23]:

- Reducing the impact of risks by developing an emergency plan if risks occur.
- Avoiding risks by changing product design.
- Accepting the risk with its consequences.
- More risk investigation so as to get more accurate information about the nature of the risk and made decisions accordingly.

4.5 Risk tracking

This process is basically meant to monitor the risks' conditions and the actions handled to deal with them. The proper risk measures are to be identified to enable risk status assessment and also the plans to reduce these risks. Tracking functions as the "watching" of management [24]. The results of tracking could be the identification of the new emerging risks that should be added to risk list, the validity of known risk solutions where risks could be eliminated from risk lists because they do not threat the project anymore, information which might give a better vision and so a better planning, implementation of emergency plan. Risk Tracking can be conducted using different software metrics. For example, Gantt charts, and gained value measures, and budget resource measures could be of much benefit in identifying and tracking risks that have differences between plans and the actual performance. Requirements churn, flaw identification proportions, and defect accumulation of work can be applied to track rework risks, risks to the quality of the submitted product, and even schedule risks [17].

4.6 Risk control

The main function of risk control is to correct the deviations from actions that were planned to face risks. As soon as risk metrics have been selected, there is nothing distinctive left for risk control. Risk control dominates project management and heavily relies on project management processes to dominate the plans assigned for risk confrontation schemes, and correcting differences amongst plans, the quick response for stirring actions, and finally the improvements of risk management processes [24].

4,7 Risk communication

The effective communication is a backbone for effective risk management. In the time that communications play a major role in facilitating interaction between the mode's elements, a higher level communications are to be considered. For a better management and handling of the risks, these risks should be well communicated between the specialized organizational levels. The parties that should be parts of the communication process include the development project and organization, the customer organization, and most importantly, the developer, the customer and, the user.

Due to the universality of communication, our approach is to deal with it as a basic part of every action taken by risk management and not as something marginal or complementary to other actions [24]. Risk communication is the core of software engineering institute's (SEI) model which asserts its importance.

5. RISK RESPONSE STRATEGIES

Generic choices for responding to project risks have been described in scientific literature such as Kliem and Ludin [13], Kendrick [25], DeMarco and Lister [26] and Frame [27]. Within the framework of these high-level choices, the specific responses could be formulated according to the project's status , the anticipated threat ,the cost of the response, and the resources needed for the response. In general, the strategies taken in response to risks usually aim to either to reduce or eliminate the probability of the risk occurrence (that is, to reduce P); undermine the impact of the risk (reduce I); or both. These strategies are usually formulated and executed in response to the new emerging risks as identified and evaluated as a controllable threat. There are four typically responses for risks as follow:

5.1 Avoidance

The main role of avoidance strategies is to prevent any negative impact that might badly affect the project that might include changing the project design in a way that there would be no chance for any risk to occur, or even to have a really influential effect on the project if it occurs. For example, the planned mission might be the "elimination" an uncertain trait to a separate stage or project where more flexible improvements could be applied to identify the requirements [28].

While avoidance strategies aim to prevent risks entirely, they can be challenging to implement without significantly altering the project's scope or design. This approach may lead to increased costs or delays, as changes to the project design can be complex and time-consuming. Additionally, avoiding risks altogether might result in missed opportunities for innovation or improvement.

5.2 Transmission

In this strategy, the responsibility of a risk is transformed to a third party. This procedure does not necessary eliminate the threat that the projects faces, it is just responsibility shifting to another person. Theoretically, this procedure suggests an agent who is fit to deal with the risk better than the current one. This shifting might have better comprehensive outcomes of the project. This strategy might be of great danger because the project threat is still present, which the chief principle has to take the responsibility for it, but the direct control is handed to the agent. The transmission strategies usually include insurance, contracts, and outside assistance. In most cases, a raise is usually paid to the agent under the title of risks raise for accepting the risk ownership. The agent is supposed to develop a certain strategy for the risk.

Transmission, or risk transfer, often involves passing responsibility to a third party, such as through insurance or contracts. However, this strategy does not eliminate the risk but rather shifts it, which can create dependency on external parties. If the third party fails to manage the risk effectively, the original project still suffers the consequences. Furthermore, the cost of transferring risk, such as premiums or fees, can be high, potentially affecting the project's budget.

5.3 Reduction

Risk reduction is one of the most promoting procedures which is planned to reduce the project threat through reducing probability/ or its expected impact prior to realizing the risk. The ultimate aim of this strategy is to manage the project in a way that risk does not take place, or if it happens, it could be contained (that is, to 'manage the threat to zero'). For example, validating the software during the development stages by testers and scripts leads to the probability of reducing post-delivery defects as well as reducing delays.

Risk reduction strategies aim to minimize the likelihood or impact of risks, but they often require significant upfront investment in time, resources, and planning. These strategies might not be entirely foolproof, as some risks can only be partially mitigated. Additionally, over-reliance on risk reduction can lead to a false sense of security, potentially causing stakeholders to underestimate residual risks.

5,4 Acceptance

Accepting a risk might comprise both active and passive strategies for facing risks. The passive response is to accept the risk as it is preferring not to take any action against it more than keeping an eye upon its status. According to Schmidt et al. [29], this response could be adopted if the risk is not that serious or low, and when the threat source is outside the project's management. However, sometimes the threat is serious but nothing can be done against it. In such a case, emergency cases could be established to deal with the case as far as it occurs. Emergencies could take the form of supplying extra financial aids or other available funds, or it could be an emergency plan which is previously prepared to deal with risk when they appear. To validate the emergency plans and maintain them is an important part of this strategy to guarantee establishing emergency plans as expected when required.

Acceptance involves acknowledging the risk and choosing to monitor it rather than taking active steps to mitigate it. This strategy is generally used when the cost of mitigation outweighs the potential impact of the risk. However, the passive nature of this approach means that if the risk materializes, the project could face significant disruptions. Emergency plans can help, but they are reactive rather than proactive, which might not be sufficient in all scenarios.

Generally speaking, the risk response strategies could be of much effect in offering general options for formulating responses against the expected risks that threaten the project. Each of these strategies requires a certain response to be planned, implemented, and reevaluated as long as the project is present where the risks nature are revealed or noticeably changed. However, because risk is still not adequately defined, these strategies are not expected to offer responses that might be applied to unexpected risks

Each risk response strategy offers distinct advantages, but they also come with limitations and challenges that must be carefully considered. A balanced approach that combines multiple strategies, tailored to the specific risks and project context, is often necessary to effectively manage risks. Continuous evaluation and adjustment of these strategies are crucial to address the dynamic nature of project risks.

6. SOFTWARE FAILURE STANDARDS

IEEE standards linked to software failure are explained below [30], these standards offer a framework for identifying and addressing potential points of failure, ensuring that risk management strategies adhere to industry benchmarks.

Aligning risk management practices with established software failure standards, such as the IEEE Software Failure Criteria, can significantly improve project outcomes by providing a consistent and industry-recognized framework for identifying and mitigating potential risks. These standards help ensure that risk management strategies are comprehensive, systematic, and adhere to best practices, which reduces the likelihood of project failures. Additionally, compliance with these standards enhances the reliability of software systems, builds stakeholder confidence, and can streamline communication across project teams by establishing a common language for discussing risks.

6.1 ***1012-2017*** *Standard*

1012-2017 standard is dedicated to deal with the system's verification and validation process, software, and hardware level. Each of the term systems, software and hardware include documentation. Verification and validation processes comprise the software product’s analysis, its assessment, its review, and testing.

6.2 ***1633-2016*** *Standard*

This standard presents ways to evaluate and expect software authenticity. It provides the needed for weighing software reliability.

6.3 ***24748-4-2016*** *Standard*

This standard describes in detail the demands concerning software life cycle process models applications. This standard also leads to the content needed in the creation of software engineering management planning report.

6.4 ***15289-2015*** *Standard*

This standard gives the precise standard or template for the content of all records created in the software life cycle. This International Standard supports ISO/IEC/IEEE 15288, ISO/IEC 12207:2008, IEEE Std 20000-1:2013, and IEEE Std 20000-2:2013.

6.5 ***730-2014*** *Standard*

Quality assurance process initiation, preparation, performing and dominating for software projects. are the duties assigned with this standard. This standard is synchronized with ISO/IEC/IEEE 12207:2008 and the information demands of ISO/IEC/IEEE 15289:2011.

6.6 ***15026-1-2014*** *Standard*

This standard describes the terms linked to assurance. Also, it furnishes the foundation for a shared understanding of assurance across user communities.

6.7 ***15026-3-2013*** *Standard*

This standard provides information regarding integrity levels with its equivalent requirements that are essential to be fulfilled to illuminate the realization of the integrity levels.

*6.8 **15026-4-2013** Standard*

This standard provides the direction for executing preferred processes, activities, and tasks for those software products that demand assurance claims for critical features.

*6.9 **29119-1-2013** Standard*

This standard describes and illustrates the ideas and glossary on which these testing standards are gathered.

*6.10 **29119-2-2013** Standard*

The standard specifies the software testing process that might happen at organizational, test administration, and active test levels. 29119-3-2013- This standard covers the templates of test documentation

*6.11 **828-2012** Standard*

This standard prepares the limited requirements for processes for Configuration Management (CM) in software engineering projects. This standard is the extension of the former configuration management standards. This standard ancestor listed only the software configuration management plan contents. Whereas this standard treats what configuration management actions are to be executed when they should be performed in the software improvement life cycle, and for doing configuration management what preparation and resources are required. Also, this standard records concerning the configuration management design content. This standard trains with ISO/IEC/IEEE 12207:2008 and ISO/IEC/IEEE 15288:2008 and remain to the terms and vocabulary in ISO/IEC/IEEE Std 24765.

*6.12 **24748-2-2012** Standard*

This standard uses ISO/IEC TR 24748- 2:2011. It offers the direction to approach ideas concerning the system, life cycle, organization and project.

*6.13 **24774-2012** Standard*

This standard defines process models. Each of them is defined by its content, format and prescription level. This standard assists to possess a unity in indicating process models.

*6.14 **26511-2012** Standard*

This standard describes the schemes to run the user documentation throughout the software development life cycle. In order to accurately control the documentation, specific features must be taken care of. The examined features can be linked to - Documentation management process, Information management process, Role of documentation team, Measurement and calculations for management control, Resource Management, Quality Management, Process Improvement, and Documentation management plan

*6.15 **15026-2-2011** Standard*

This standard establishes the least claims for the organization and contents of an assurance case to improve its compatibility. Also, the standard serves to aid communication amongst stakeholder, and help engineering judgments of assurance cases. Assurance cases are generally formed to hold claims in features like protection, compatibility, maintainability, anthropological factors, and operability.

*6.16 **24748-1-2011** Standard*

This standard grants direction towards software life cycle concepts, its explanation, meaning, and results. It leads to choosing a suitable process model for promoting a software project.

*6.17 **26512-2011** Standard*

This standard defines the way to support the different users to obtain or provide software user documentation as an element of the software development life cycle. This standard more grants support to describe the process of documentation from acquirer's and supplier's view.

*6.18 **29148-2011** Standard*

This standard is the substitution of IEEE 830-1998, IEEE 1233-1998, IEEE 1362-1998. ISO/IEC/IEEE 29148:2011. This standard develops the processes associated with software demand engineering.

*6.19 **42010-2011** Standard*

This standard treats the design description concerning production, interpretation and sustainment of systems. In particular, the standard develops the architecture perspectives, structures, and general methods for representing a structure.

6.20 ***1517-2010*** *Standard*

This is a frame that increases the IEEE Std 12207(TM)-2008 and joins the methodical practice of reuse. It allows a system to be generated from reusable assets.

6.21 ***26513-2010*** *Standard*

This standard provides necessity towards testing and evaluating of software user documentation as a segment of the software development life cycle. It gives detail means to apply in testing and reviewing the user's documentation.

6.22 ***26514-2010*** *Standard*

This standard provides necessity towards composing and developing of software user documentation as a part of the software development life cycle. It develops the documentation process from the developer's viewpoint.

6.23 ***1016-2009*** *Standard*

This standard specifies the necessary information for software design descriptions. This software design description depicts software design that will be utilized for transmitting information about design to its related stakeholders. This standard is suitable for automated databases and design depiction languages. More precisely this standard can be employed for hand-operated records and other means of descriptions.

6.24 ***1044-2009*** *Standard*

This standard donates a patterned path to the organization of software discrepancies within the project lifecycle. Classification serves to decrease the risks of deficit insertion or to improve the possibility of early defect detection.

6.25 ***16326-2009*** *Standard*

This standard defines content for managing projects.

6.26 ***1028-2008*** *Standard*

This standard deals with the representation of five different revisions that might be needed through software development life cycle. The different review samples are management reviews, technical reviews, inspections, walkthroughs, and audits.

6.27 ***14764-2006*** *Standard*

This standard presents a guideline for methods to maintain and execute software maintenance exercises.

6.28 ***16085-2006*** *Standard*

This standard describes the process for handling risk in the software development life cycle.

6.29 ***1061-1998*** *Standard*

This methodology is practised for creating quality demands. Furthermore, the methodology is employed to classify, execute, interpret and certify the quality metric correlated to process and product.

Each of the IEEE standards for software engineering is examined for factors and sub-factors which might lead to fluctuating situational contexts. Based on the interpretation of the chosen standards, it is affirmed that standards hold different focus areas.

Incorporating machine learning (ML) methods into risk assessment processes can significantly enhance the effectiveness of risk management strategies [31 - 35]. ML models are used to predict system risks by categorizing each software project as a "fail" or "success" based on specific constraints. These models classify data from a training set and predict outcomes for new data, facilitating informed decision-making in new situations. This approach, known as supervised learning, allows organizations to develop more precise classification procedures and apply them to real-time risk assessments.

7. CONCLUSION

Effective risk management is crucial for the success and stability of software projects, given their inherent complexity and liability to failure. This paper explores a literature on risk analysis and management, highlighting the importance of understanding and addressing risks to prevent project failure. The paper highlights that classifying risks according to their duration, impact, and source can significantly improve effectiveness of risk management strategies. A systematic approach to risk management, including identification, classification, analysis, planning, tracking, control, and communication, offers a robust framework for mitigating potential threats and minimizing their impact. Various risk response strategies, such as avoidance, transfer, reduction, and acceptance, provide diverse methods for managing risks based on their nature and severity. By aligning these strategies with established standards like the IEEE Software Failure Criteria, organizations can ensure compliance with industry benchmarks and enhance the reliability of their risk management practices. In conclusion, effective risk management is fundamental to the success of software projects. Through a structured approach to risk assessment and the application of appropriate response strategies, organizations can navigate uncertainties more effectively, improve project outcomes, and achieve their objectives with greater confidence.

REFERENCES

[1] Kalinowski, M., Spínola, R. O., Conte, T., Prikladnicki, R., Méndez Fernández, D., & Wagner, S. (2015). Towards building knowledge on causes of critical requirements engineering problems.

[2] Bourque, P., & Fairley, R. E. (2014). Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press.

[3] Team, C. M. M. I. (2002). CMMI for software engineering, version 1.1, staged representation (CMMI-SW, V1. 1, Staged).

[4] Thayer, R. H. (2002). Software system engineering: A tutorial. Computer, 35(4), 68-73.

[5] Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave Macmillan.

[6] Sommerville, I., & Prechelt, L. (2004). Verification and validation. Software Engineering, 7.

[7] McConnell, S. (2004). Code complete. Pearson Education.

[8] Hopkin, P. (2018). Fundamentals of risk management: understanding, evaluating and implementing effective risk management. Kogan Page Publishers.

[9] Boehm, B. W. (1991). Software risk management: principles and practices. IEEE software, 8(1), 32-41.

[10] Sadgrove, K. (2016). The complete guide to business risk management. Routledge.

[11] Charette, R. N. (2017). Why Software Fails. 2005. URL: http://spectrum. ieee. org/computing/software/why-software-fails (hämtad 2016-05-09).

[12] Winch, G., & Leiringer, R. (2016). Owner project capabilities for infrastructure development: A review and development of the “strong owner” concept. International Journal of Project Management, 34(2), 271-281.

[13] Kliem, R. L., & Ludin, I. S. (2019). Reducing project risk. Routledge.

[14] Chapman, C., & Ward, S. (2004). Why risk efficiency is a key aspect of best practice projects. International Journal of Project Management, 22(8), 619-632.

[15] Boehm, B., Lane, J. A., Koolmanojwong, S., & Turner, R. (2014). The incremental commitment spiral model: Principles and practices for successful systems and software. Addison-Wesley Professional.

[16] Jiang, J. J., Klein, G., & Discenza, R. (2001). Information system success as impacted by risks and development strategies. IEEE transactions on Engineering Management, 48(1), 46-55.

[17] Westfall, L. (2000). Software risk management. In Annual Quality Congress Proceedings-American Society for Quality Control (pp. 32-39). ASQ; 1999.

[18] Giannakis, M., & Papadopoulos, T. (2016). Supply chain sustainability: A risk management approach. International Journal of Production Economics, 171, 455-470.

[19] Tiwana, A., & Keil, M. (2004). The one-minute risk assessment tool. Communications of the ACM, 47(11), 73-77.

[20] Dey, P. K. (2012). Project risk management using multiple criteria decision-making technique and decision tree analysis: a case study of Indian oil refinery. Production Planning & Control, 23(12), 903-921.

[21] Wanderley, M., Menezes Jr, J., Gusmao, C., & Lima, F. (2015). Proposal of risk management metrics for multiple project software development. Procedia Computer Science, 64, 1001-1009.

[22] Lundgren, R. E., & McMakin, A. H. (2018). Risk communication: A handbook for communicating environmental, safety, and health risks. John Wiley & Sons.

[23] Twigg, J. (2015). Disaster risk reduction. Overseas Development Institute, Humanitarian Policy Group.

[24] Haimes, Y. Y. (2015). Risk modeling, assessment, and management. John Wiley & Sons.

[25] Kendrick, T. (2015). Identifying and managing project risk: essential tools for failure-proofing

your project. Amacom.

[26] DeMarco, T., & Lister, T. (2003). Risk management during requirements. IEEE software, 20(5), 99-101.

[27] Frame, J. D. (2003). Managing risk in organizations: A guide for managers. John Wiley & Sons.

[28] Routledge. Boehm, B., & Turner, R. (2003). Using risk to balance agile and plan-driven methods. Computer, 36(6), 57-66.

[29] Schmidt, R., Lyytinen, K., Keil, M., &Cule, P. (2001).Identifying software project risks: An international Delphi study. Journal of management information systems, 17(4), 5-36.

[30] Khan, Huma Hayat, and Muhammad Noman Malik. "Software standards and software failures: a review with the perspective of varying situational contexts." IEEE access 5 (2017): 17501-17513.

[31] Ibraigheeth, Mohammad, and Aws Ismail. "Software project risk assessment using machine learning approaches Software project risk assessment using machine learning approaches." Am J Multidiscip Res Dev. https://www. researchgate. net/publication/358564485%0ASoftware (2022).

[32] Ibraigheeth, Mohammad Ahmad, and Syed Abdullah Fadzli. "Fuzzy Logic Driven Expert System for the Assessment of Software Projects Risk." International Journal of Advanced Computer Science and Applications 10.2 (2019).

[33] Ibraigheeth, M. A., & Fadzli, S. A. (2020, October). Software project failures prediction using logistic regression modeling. In 2020 2nd International Conference on Computer and Information Sciences (ICCIS) (pp. 1-5). IEEE.

[34] Ibraigheeth, M., & Fadzli, S. A. (2018). Software reliability prediction in various software development stages. Journal of theoretical and applied information technology, 96(7).

[35] Ibraigheeth, M. A., Abu Eid, A. I., Alsariera, Y. A., Awwad, W. F., & Nawaz, M. (2024). A New Weighted Ensemble Model to Improve the Performance of Software Project Failure Prediction. International Journal of Advanced Computer Science & Applications, 15(2).